Few-Shot Class-Incremental Learning from an Open-Set Perspective

Can Peng[1], Kun Zhao[2], Tianren Wang[1], Meng Li[1], and Brian C. Lovell[1]

[1] The University of Queensland, Brisbane, QLD, Australia
[2] Sullivan Nicolaides Pathology, Australia
can.peng@uq.net.au, kun_zhao@snp.com.au, tianren.wang@uq.net.au, meng.li6@uq.net.au, lovell@itee.uq.edu.au

Abstract. The continual appearance of new objects in the visual world poses considerable challenges for current deep learning methods in real-world deployments. The challenge of new task learning is often exacerbated by the scarcity of data for the new categories due to rarity or cost. Here we explore the important task of Few-Shot Class-Incremental Learning (FSCIL) and its extreme data scarcity condition of one-shot. An ideal FSCIL model needs to perform well on all classes, regardless of their presentation order or paucity of data. It also needs to be robust to open-set real-world conditions and be easily adapted to the new tasks that always arise in the field. In this paper, we first reevaluate the current task setting and propose a more comprehensive and practical setting for the FSCIL task. Then, inspired by the similarity of the goals for FSCIL and modern face recognition systems, we propose our method — Augmented Angular Loss Incremental Classification or ALICE. In ALICE, instead of the commonly used cross-entropy loss, we propose to use the angular penalty loss to obtain well-clustered features. As the obtained features not only need to be compactly clustered but also diverse enough to maintain generalization for future incremental classes, we further discuss how class augmentation, data augmentation, and data balancing affect classification performance. Experiments on benchmark datasets, including CIFAR100, miniImageNet, and CUB200, demonstrate the improved performance of ALICE over the state-of-the-art FSCIL methods. Code is available at https://github.com/CanPeng123/FSCIL_ALICE.

Keywords: Few Shot, One Shot, Incremental Learning, Classification

1 Introduction

In recent years, the computer vision community has witnessed astonishing performance breakthroughs in many traditional vision tasks. These breakthroughs are mainly due to the emergence of deep learning models and algorithms, publicly available large data sets for training, and powerful GPU computing devices. Despite their popularity, current deep learning techniques mostly rely on large-scale supervised data to train accurate models. A deep neural network (DNN) with tens of thousands of parameters cannot be easily adapted to a new task by

training on just a few examples. In addition, conventional deep learning models lack the capability of preserving previous knowledge while adapting to new tasks. When a neural network is fine-tuned to learn a new task, its performance on previously trained tasks will significantly deteriorate, a problem known as catastrophic forgetting [8,16]. Exploring the fast learning and memorizing capability of deep learning models is an important step toward improving their practical application ability.

In this paper, we tackle this significant research direction — Few-Shot Class-Incremental Learning (FSCIL). FSCIL requires the trained model to not only quickly adapt to continually arriving new tasks, but also to retain the old knowledge about previously learned tasks. Considering real-life application, an ideal FSCIL model needs to have the following characteristics: 1) The model needs to perform well on all classes equally, no matter what the training presentation sequence is; and 2) the model needs to be robust to extreme data scarcity, such as the one-shot scenario. However, current SOTA methods mainly use sole class-wise average accuracy to evaluate the model performance which cannot assess whether there is a prediction bias due to class imbalance and data imbalance. As there are normally more base classes than incremental classes and only limited data is provided for each incremental class, prediction bias towards base classes can easily happen. In addition, current SOTA methods rarely consider the extreme one-shot setting which can happen in the real world due to incremental data collection and rare data types. A well-established task setup is a cornerstone for the development of this task since an improper task setup will misguide the method design and lead to methods with limited application. Thus, before designing our method, we reformulate the setup for the FSCIL task.

Considering the paucity of incremental session data and the absence of old session data, we think the feature extractor trained on the base session should not be limited to extracting discriminative features for the base categories. The ability of representing new unseen samples from future novel classes is also critical. On the one hand, we are motivated by the similarity between FSCIL and face recognition tasks. The face recognition system learns to distinguish and recognize new faces quickly via its deep metric learning framework. The capability of handling new identities without the need for retraining is a major achievement of modern face recognition methods and is also what the FSCIL task desires. On the other hand, we are motivated by the intuitive connection between FSCIL and data augmentation. Data augmentation focuses on improving the generalization of a DNN. The capability of extracting diverse features that is transferable across base and incremental classes is important for the FSCIL task. Hence in this work, we adopt some ideas from both modern face recognition and data augmentation to design our method.

The contributions of this paper are: **(1)** We reevaluate the current benchmark task settings of FSCIL and propose additional experimental settings and evaluation metrics to more comprehensively assess the capability of FSCIL methods. **(2)** We solve the FSCIL task from a new perspective of the open-set problem. We analyze the angular penalty loss from face recognition and adapt it to

FSCIL to improve the discrimination of the model. **(3)** We further analyze how data processing, such as class augmentation, data augmentation, and balanced data embedding affect FSCIL performance and aim to improve the generalization of the model. **(4)** Significant improvements on three benchmark datasets, CIFAR100, miniImageNet, and CUB200, demonstrate the effectiveness of our method against SOTA methods.

2 Related Work

Few-shot Class-incremental Learning. The FSCIL task is a newly emerged challenge evolved from class-incremental learning [17,1,11]. Once established, the research community has spent much effort developing algorithms for this important FSCIL task. For SOTA FSCIL methods, after base session training, some update the backbone [19,4,25,7] and some freeze the backbone [23,27,5]. Backbone updating methods commonly use the knowledge distillation [10] technique to preserve the old knowledge. Knowledge distillation relies on having sufficient data to simulate the input-output function of the old model. To adapt knowledge distillation to FSCIL, these methods store old exemplars, require a complex updating scheme for each incremental task, or are incapable of extreme data scarcity conditions such as 1-shot. However, high performance and flexible operation are both important for real-world applications. Also, storing old exemplars is undesirable due to memory restrictions. In addition, the backbone network has a large number of parameters despite there being extremely limited new task data. The large imbalance between parameters and data causes the backbone updating methods to normally show lower performance than backbone freezing methods under the same experimental setup.

On the contrary, freezing the backbone network is a good choice to well balance not only the real-life application requirements but also the stability and plasticity trade-off. This backbone freezing strategy decouples the learning of representations and classifiers to avoid overfitting and catastrophic forgetting in the representations. Also, the fundamental feature characteristics are similar for many objects, so features learned from the base session can be readopted for recognizing new classes. Our method belongs to the backbone freezing type of methods. Although this decoupling strategy has been explored by Zhang *et al.* [23], their method focuses on designing a discriminative classifier. On the contrary, we focus on feature distribution, since this is a cornerstone of robust classification performance. Last but not the least, a good FSCIL method needs to perform equally well on all the classes no matter whether they are base or incremental classes. This is a problem for the current backbone freezing type of methods that their good overall accuracy is mainly derived from the base session. In this paper, we target on proposing an FSCIL method that takes advantage of decoupling representation and classification via backbone freezing, and at the same time, solves the side effect of prediction bias.

Deep Metric Learning. Deep metric learning is commonly used for face recognition tasks. Inspired by the relation between normalized weights on the last fully

connected layer and class centers, Liu *et al.* proposed SphereFace [14] which uses an angular margin penalty to enforce extra intra-class compactness and inter-class discrepancy. Following SphereFace, CosFace [21] and ArcFace [6] were proposed to reduce the complex loss calculation and make the training procedure more stable. There are many similarities between face recognition and FSCIL tasks: 1) both tasks are open-set object recognition tasks that need to classify a large amount of continually arriving new objects (classes/face identities); 2) both tasks are provided with unbalanced data; and 3) both tasks require fast adaptation on new objects as well as maintaining performance on old objects. Inspired by these similarities, in this paper, we try to solve the FSCIL task from a new perspective of the open-set problem. We adopt the idea of angular penalty loss from face recognition to the more general problem of object recognition.

As real-world classification problems typically exhibit class imbalance or long-tailed data distribution, some methods have explored deep metric learning for incremental and long-tail tasks [22,15,12]. However, these methods normally assume sufficient data is available which is a different setting from FSCIL. Most FSCIL methods solve this problem from the perspective of either incremental learning (advanced knowledge distillation) [4,25,7] or few-shot learning (freezing backbone and evolving prototypes) [23,5,27]. We follow the proposal of freezing the backbone network to decouple the learning of representations and classifiers. However, different from current backbone freezing type of methods that maintain the incremental learning ability by evolving classification prototypes, we focus on improving the transfer capability of the feature extractor.

3 Problem Formulation

FSCIL task comprises a base task with sufficient training data and multiple incremental tasks with limited training data. During the learning of each new task, only the data for the current task is available and the model is required to learn this new task information whilst retaining old task knowledge.

To be specific, assume an M-step FSCIL task. Let $\{D^0_{train}, D^1_{train}, ..., D^m_{train}\}$ and $\{D^0_{test}, D^1_{test}, ..., D^m_{test}\}$ denotes the training and testing data for sessions $\{0, 1, ..., m\}$, respectively. For session i, it has training data D^i_{train} with the corresponding label space of C^i. Training data from different sessions have no overlapped classes, so when $i \neq j$, $C^i \cap C^j = \varnothing$. During testing, the model will be evaluated on all seen classes so far, so for session i, its testing data D^i_{test} has the corresponding label space of $C^0 \cup C^1 ... \cup C^i$. In addition, for the base session ($i = 0$), a sufficient amount of training data is provided and for the following incremental sessions ($i > 0$), only a limited amount of data is provided.

Most papers about FSCIL [4,25,7,23,27,5] follow the task setting proposed by Tao *et al.* [19]. As FSCIL focuses on mimicking real-life situations, we think some aspects of the current benchmark experimental protocol are not sufficient to evaluate the efficiency of an FSCIL method. Thus, before proposing our method, we propose a more comprehensive and practical setup for the FSCIL task.

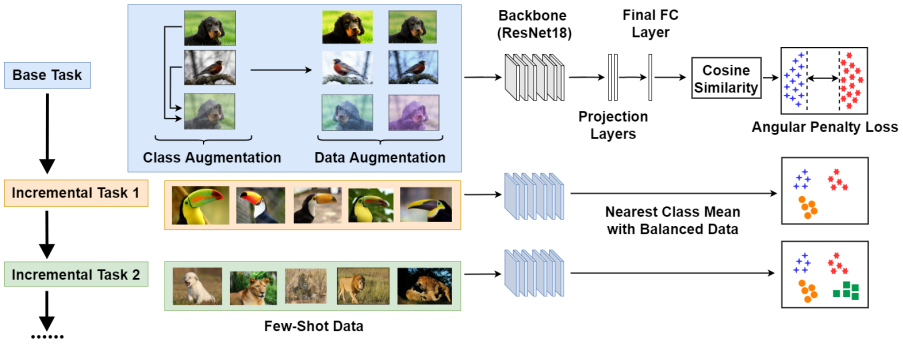

Fig. 1: The framework of our proposed method. On the one hand, with sufficient base task data available, angular penalty loss, class augmentation, and data augmentation are utilized to obtain a general open-set feature extractor. On the other hand, as only limited incremental task data is available, the few-shot new class data and the carefully chosen same number of base class data are utilized to generate the balanced class-wise prototypes. Nearest class mean and cosine similarity are adopted to do the final classification.

Number of Few-Shot Data. Current benchmark experiments are performed with 5-shot, 10-shot, or more data being available for each incremental step. The extreme data scarcity condition of 1-shot which can easily happen in the real world due to extremely scarce data type is rarely considered.

Evaluation Metric. Current benchmark evaluation metrics mainly use class-wise average accuracy to evaluate the performance of an FSCIL model. As there are normally more base classes than incremental new classes, using average accuracy cannot indicate if there is a prediction bias between base and incremental classes. A method cannot be regarded as a good FSCIL method if its good performance is mainly determined by the base class performance.

Dataset. The similarity between base classes and new classes will strongly affect model performance since the high re-usability of base features such as fine-grained datasets will naturally reduce the challenge of catastrophic forgetting. An optimal FSCIL model needs to not only perform well on high-distributional-match fine-grained datasets but also on low-distributional-match datasets.

To sum up, to comprehensively simulate the real-world FSCIL condition and evaluate the robustness of an FSCIL method, we consider both benchmark 5-shot and 1-shot settings. Also, for the evaluation metric, we propose to use both average accuracy and harmonic accuracy to evaluate not only the overall performance but also the performance balance between base and incremental classes. In addition, we perform experiments on both general (CIFAR100 and mini-ImageNet) and fine-grained (CUB200) datasets to remove the possible performance benefit due to high similarity between base and incremental classes.

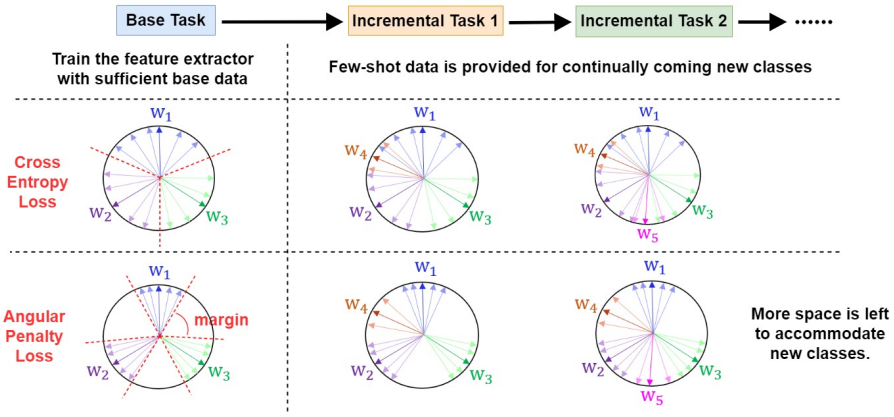

Fig. 2: An illustration of feature distributions of a cross-entropy loss trained model and an angular penalty loss trained model. The light color arrows represent examples of different class features on the latent feature space. The dark color arrows represent the average feature prototype of corresponding classes. Angular penalty loss provides more compact intra-class clustering and wider inter-class separation than cross-entropy loss. Compact clustering leaves more room on the latent feature space to accommodate the new classes.

4 Methodology

In this section, we propose the FSCIL method ALICE using angular penalty, class and data augmentation, and data balancing. First, for the base session, we apply the angular penalty loss to train the feature extractor to obtain compact intra-class clustering and wide inter-class separation. Class augmentation and data augmentation are also adopted to improve the generalization of the feature extractor. Then, for the incremental sessions, specifically chosen balanced data are utilized to generate prototypes for each class. Nearest class mean and cosine similarity are combined to perform the classification. Figure 1 demonstrates the framework of our method.

4.1 Angular Penalty

Under the FSCIL setting, we want to obtain a feature extractor which can rapidly adapt to continually coming new tasks, as well as be stable to overcome catastrophic forgetting for the previously learned tasks. Thus, we want to use a loss function that: 1) minimizes the distance between intra-class feature vectors, and 2) maximizes the distance between inter-class feature vectors. The compact intra-class clustering and wide inter-class separation will leave more room in the latent feature space for the incrementally arriving new classes and hence lead to better open-set classification. Figure 2 illustrates an example. As many innovative angular penalty losses have been explored and proposed for face recognition studies

[21,6] and considering the similarity between FSCIL and face recognition tasks, we adapt the cosFace penalty strategy [21] to FSCIL training.

First, we use cosine similarity as the distance metric to measure data similarity and compute scores. It has two effects: 1) it makes training focus on the angles between normalized features instead of absolute distance in the latent feature space, and 2) the normalized weight parameters of the fully connected layer can be regarded as the center of each category. To calculate cosine similarity in the final fully connected layer, we fix the bias to 0 for simplicity. Then the data prediction procedure can be written as:

$$f = \mathcal{F}(x) \tag{1}$$

$$y_i = W_i^T f = \|W_i\|\|f\| \cos(\theta_i) = \cos(\theta_i),$$

$$\|W_i\| = \|f\| = 1 \tag{2}$$

where f is the feature obtained from the input image x through the feature extractor $\mathcal{F}$. The feature f and the weight parameter W_i are normalized by $\ell 2$ normalization, so the magnitude is 1. The quantity y_i is the calculated cosine similarity between the feature f and the weight parameter W_i for class i. It measures the angular similarity of image x towards class i which indicates the likelihood that image x belongs to class i.

Normally, the cosine similarity prediction is used with cross-entropy loss to separate features from different classes by maximizing the probability of the ground-truth class. The loss function is:

$$\begin{aligned} L &= -\frac{1}{N}\sum_{j=1}^{N}\log(p_j) = -\frac{1}{N}\sum_{j=1}^{N}\log(\frac{e^{y_j}}{\sum_{i=1}^{C} e^{y_i}}), \\ &= -\frac{1}{N}\sum_{j=1}^{N}\log(\frac{e^{\|W_j\|\|f\|\cos(\theta_j)}}{\sum_{i=1}^{C} e^{\|W_i\|\|f\|\cos(\theta_i)}}), \\ &= -\frac{1}{N}\sum_{j=1}^{N}\log(\frac{e^{\cos(\theta_j)}}{\sum_{i=1}^{C} e^{\cos(\theta_i)}}) \end{aligned} \tag{3}$$

where N is the number of training images and C is the number of classes. The quantity p_j describes the softmax probability for image j. The quantity y_j describes the cosine similarity towards its ground truth class for image j.

To make features better clustered, inspired by cosFace [21], a cosine margin m is introduced to the classification boundary. With the help of the extra margin, the intra-class features become more compactly clustered and the inter-class features become more widely separated. Following cosFace, we also re-scale the normalized feature by a preset scale factor s. The loss function is:

$$L_{AP} = -\frac{1}{N}\sum_{j=1}^{N}\log(\frac{e^{s(\cos(\theta_j)-m)}}{e^{s(\cos(\theta_j)-m)} + \sum_{i\neq j} e^{s\cos(\theta_i)}}) \tag{4}$$

The scale factor s is set to 30 and the cosine margin m is set to 0.4 for all experiments.

4.2 Augmented Training

Diverse and transferable representation is the key for open-set problems. Exposure to a large number of classes is one way to obtain such kind of feature extractors. To this end, a simple and effective method is to introduce auxiliary classes. Inspired by Mixup [24] and IL2A [26], we randomly combine pairs of different class examples from the base session data to synthesize auxiliary new class data. The new class data generating function is:

$$x_k = \lambda x_i + (1 - \lambda) x_j \tag{5}$$

where x_i and x_j are two training samples from two different classes i and j randomly picked from the C base session classes. λ is the interpolation coefficient. x_k is the generated new class data. Figure 3 shows an example. In our experiments, following IL2A [26], we restrict λ to be a randomly chosen value between $[0.4, 0.6]$ to reduce the overlap between the augmented and original classes. For a C-class classification task, by pair combination, we will generate $(C \times (C-1)/2)$ new classes, so the original C-class classification task now becomes a $(C + C \times (C-1)/2)$-class classification task.

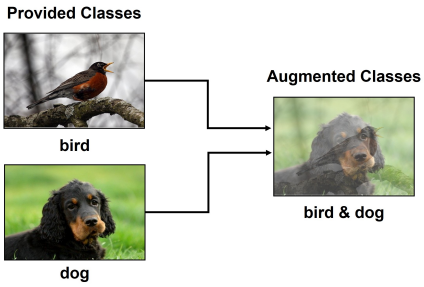

Fig. 3: An example of class augmentation. Auxiliary new class data is generated by interpolating two different class samples from base session data.

Exposure to various image conditions during training is also a good method to obtain a general feature extractor. Inspired by self-supervised learning [2,3], we use two augmentations of each image to enhance training data diversity. Figure 1 shows the augmentation procedure. During training, for each input image, we randomly generate two augmentations from a set of preset transformation strategies. For the utilized transformation methods, we randomly apply resized crop, horizontal flip, color jitter, and grayscale. Then both transformed data are sent to the backbone network. The losses from two sets of augmentation are averaged and back-propagated to update model parameters. In addition, to avoid the feature extractor over-specialize to base session data, following SimCLR [2], we utilize extra projection layers before the final fully connected layer. By leveraging the nonlinear projection head, more information can be formed and maintained in the feature extractor.

4.3 Balanced Testing

After base session training, the projection head and the augmented classification head are discarded. Only the feature extractor is left and it is frozen to avoid both overfitting and catastrophic forgetting. During testing, nearest class mean and cosine similarity are utilized to do the classification. As there is only limited

data provided for each incremental session, to alleviate the possible prediction bias due to data imbalance, we use the same amount of few-shot data as the following incremental steps to generate the base class prototypes. To select suitable examples, we first use all base session data to calculate the class-wise mean for each base class. Then the required few-shot amount of data which has the smallest cosine distance with the calculated mean is used to generate the final prototype for each base session class.

4.4 Harmonic Accuracy

For the evaluation metric, current SOTA methods generally report the class-wise average accuracy. However, we argue that the class-wise average accuracy is not enough to evaluate the performance of an FSCIL method, since the number of classes from the base session is often a large fraction of the total number of classes. Following the experimental settings on benchmark papers, for CIFAR100 [13] and miniImageNet [18], 60 out of 100 (60%) categories are used as base classes. For CUB200 [20], 100 out of 200 (50%) categories are used as base classes. A model with good performance on the base session and poor performance on the following incremental sessions can still have a good average accuracy due to the high ratio of base classes to the overall classes. For example, with 60 base classes, on one step of incremental learning 5 classes, an algorithm that shows 100% accuracy on base classes with 0% on incremental classes would be rated 92.3% using average accuracy, yet it would have demonstrated no learning on the new task. To compensate for this deficiency of average accuracy, we adapt the harmonic accuracy metric that requires well-balanced performance across both base and incremental classes. The formula for harmonic accuracy (A_h) is:

$$A_h = \frac{2 \times A_b \times A_i}{A_b + A_i} \tag{6}$$

where A_b is the average accuracy for base session classes and A_i is the average accuracy for the following incremental session classes. In the simple example above, the harmonic accuracy would be 0% which is much more appropriate as the network has indeed learned nothing at all. An ideal balanced FSCIL classifier will have equally high performance on both average accuracy and harmonic accuracy. If a model has good average accuracy but poor harmonic accuracy, this means that its good performance is mainly due to performance on the base session classes and the model has poor incremental learning capability overall.

5 Experiments

5.1 Dataset and Evaluation Metric

We use three benchmark datasets CIFAR100 [13], miniImageNet [18] and Caltech-UCSD Birds-200-2011 (CUB200) [20] for our experiments. CIFAR100 contains 100 classes with 600 images per class, 500 for training and 100 for testing. Each image has a size of 32×32 pixels. MiniImageNet also contains 100 classes with

600 images per class, 500 for training and 100 for testing. Each image has a size of 84 × 84 pixels. CUB200 is a fine-grained image classification dataset. It contains 200 classes of different species of birds with 5994 training images and 5794 testing images. Each image has a size of 224 × 224 pixels.

As mentioned in section 3, to comprehensively evaluate an FSCIL method, we follow the benchmark 5-shot setting and also perform an additional 1-shot setting. For experiments on CIFAR100 and miniImageNet, the 8-step 5-way 5-short and 8-step 5-way 1-short incremental settings are used. In this protocol, 60 classes are used as base classes with all training data provided; then 40 classes are used as incremental classes with 5-shot or 1-shot training data provided in a 5-way manner in 8 steps. For experiments on CUB200, 10-step 10-way 5-shot and 10-step 10-way 1-shot settings are used. 100 classes are used as base classes and the remaining 100 classes are used as incremental classes with 5-shot or 1-shot training data provided in a 10-way manner in 10 steps. To make the evaluation comprehensive and fair, we report both average accuracy and harmonic accuracy.

5.2 Implementation Details

For our experiments, we use ResNet18 [9] as the backbone network. We implement the projection head as a two-layer MLP with a hidden feature size of 2048 and ReLU as the activation function. Our method is built with PyTorch library and SGD with momentum is used for optimization. The initial learning rate is set to 0.01 for CIFAR100 and miniImageNet dataset training, and 0.001 for CUB200 dataset training. Following the settings on [19,23], models for CIFAR100 and miniImageNet are trained from scratch, and models for CUB200 are initialized by an ImageNet pretrained model. When class augmentation is not applied, a batch size of 512 is used for training. When class augmentation is used, we use a batch size of 128 for CIFAR100 and a batch size of 64 for miniImageNet. The experimental results for CEC [23] are reproduced by their publicly available source code.

5.3 Comparison with the State-of-the-art Methods

We compare our method with the SOTA methods [17,1,11,19,4,5,23] on three datasets. According to Figure 4, for experiments on CIFAR100 and miniImageNet dataset, under both 8-step 5-way 5-shot and 1-shot settings, our method achieves the highest class-wise accuracy over all the sessions. Also, on both datasets, our ALICE method shows much higher harmonic accuracy on all sessions compared to the SOTA CEC [23] method. The high harmonic accuracy proves that our method can largely alleviate the prediction bias problem. To be more specific, in CIFAR100, for the 5-shot (1-shot) setting, in the last session, we get 54.1% (47.5%) average accuracy and 50.6% (26.5%) harmonic accuracy which is 6.0% (2.7%) and 19.3% (13.5%) higher than the CEC method, respectively. In miniImageNet, for the 5-shot (1-shot) setting, in the last session, we get 55.7% (48.6%) average accuracy and 50.9% (27.1%) harmonic accuracy which is 8.5% (4.9%) and 22.8% (19.3%) higher than the CEC method, respectively.

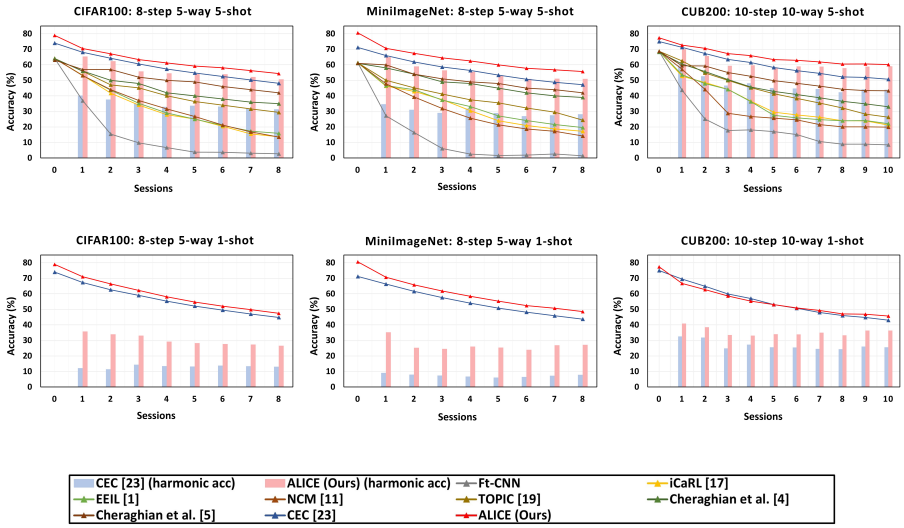

Fig. 4: Comparison with SOTA methods under both 5-shot and 1-shot incremental settings on CIFAR100, miniImageNet, and CUB200 dataset. The line chart represents average accuracy and the histogram represents harmonic accuracy. Our method outperforms SOTA works with significant performance advantages.

For experiments on the CUB200 dataset, we find that applying class augmentation will deteriorate the model performance. As CUB200 is a fine-grained dataset, the feature extractor needs to focus on learning tiny differences between categories. However, class augmentation targets obscuring the class difference and forcing the feature extractor to focus on general features. It is a good augmentation strategy to extract transferable features for general FSCIL tasks but will adversely obscure the class boundaries for fine-grained FSCIL tasks. Thus, for experiments on CUB200, we do not use class augmentation and train the feature extractor only by angular penalty and data augmentation. According to Figure 4, our method outperforms all SOTA methods by a large margin on the 5-shot setting. This proves that for fine-grained classification where the reusability of features is high, angular penalty and data augmentation is enough to obtain a robust open-set feature extractor. Under the 1-shot setting, we get similar average accuracy as CEC, since both of us freeze the backbone network after base session training to avoid catastrophic forgetting. When considering incremental class performance, our method can better adapt to new classes and obtain much higher harmonic accuracy than the CEC method.

Besides, we also compare the confusion matrices produced by CEC and our method after the last incremental session. The results are shown in Figure 5. Compared with CEC, our method produces a more balanced base and incremental class performance, especially under 5-shot settings. When under 1-shot

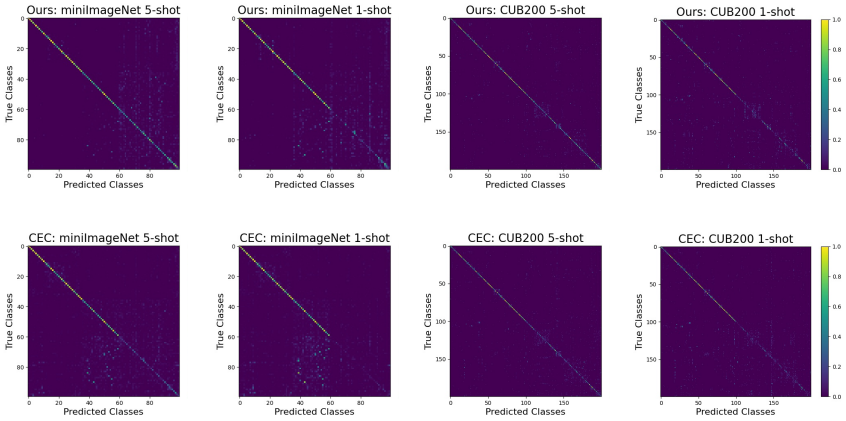

Fig. 5: Comparison of the confusion matrices produced by CEC and our method on the last incremental session for 5-shot and 1-shot incremental experiments on miniImageNet and CUB200 dataset.

settings, although our method can outperform the CEC method, the prediction bias towards base classes still exists. This is because the 1-shot setting is the most extreme FSCIL setting due to maximal data scarcity and data imbalance. We will focus on solving this problem in future work.

5.4 Ablation Study

To validate the effectiveness of each part of our method, we perform an ablation study on the CIFAR100 dataset under the 8-step 5-way 5-shot setting. Table 1 shows the experimental results for the ablation study. When balanced data are used for prototype generation, compared with the cross-entropy loss trained model, in the last incremental step, the angular penalty loss trained model provides 3.3% average accuracy improvement. But the harmonic accuracy is 6.2% lower. This means that solely using angular penalty loss will make the feature extractor over-specialize to the base session data and lose its generalization performance. The high average accuracy produced by the angular penalty loss trained model with all data used for prototype generation also shows the over-specialization. Their good average performance is mainly due to the base classes since at the first several sessions, the ratio of base classes among all classes is high. To compensate for the loss of generalization, projection layers are utilized to help the feature extractor maintain more information. With the help of the projection layers, in the last incremental session, the average accuracy remains unchanged but the harmonic accuracy is increased from 25.2% to 43.8% which is 12.4% higher than the cross-entropy loss trained model. Then, when two transformations of each input image are utilized for loss calculation, the average accuracy (harmonic accuracy) is 1.8% (3.6%) increased in the last

Table 1: Ablation study on CIFAR100 under the 8-step 5-way 5-shot setting.

loss type	class aug	data aug	project layer	balanced data	0	1	2	3	4	5	6	7	8
					class-wise average accuracy								
cross entropy	✗	✗	✗	✗	74.2	67.4	63.4	59.4	55.9	53.2	51.2	49.0	46.9
	✗	✗	✗	✓	74.2	65.4	61.6	57.7	54.5	52.1	49.9	47.9	46.2
angular penalty	✗	✗	✗	✗	76.9	**72.9**	**68.2**	**64.1**	60.3	57.0	54.3	51.9	49.7
	✗	✗	✗	✓	76.9	72.8	68.0	63.8	60.2	56.8	54.1	51.8	49.5
	✗	✗	✓	✓	74.2	67.1	63.7	59.9	56.8	54.1	52.8	51.1	49.5
	✗	✓	✓	✓	75.6	68.2	64.2	60.3	57.9	55.6	54.7	53.1	51.3
	✓	✓	✓	✓	**79.0**	70.5	67.1	63.4	**61.2**	**59.2**	**58.1**	**56.3**	**54.1**
					harmonic accuracy								
cross entropy	✗	✗	✗	✗	-	36.5	32.1	29.4	27.1	27.2	28.3	27.2	27.4
	✗	✗	✗	✓	-	45.5	37.8	34.7	32.4	32.8	32.1	30.8	31.4
angular penalty	✗	✗	✗	✗	-	34.0	28.2	26.4	23.6	23.3	22.6	22.1	22.3
	✗	✗	✗	✓	-	40.4	32.8	29.7	27.5	26.2	25.4	24.6	25.2
	✗	✗	✓	✓	-	58.9	57.2	50.5	47.9	46.4	46.4	45.0	43.8
	✗	✓	✓	✓	-	65.0	60.0	52.2	50.9	49.6	50.1	48.6	47.4
	✓	✓	✓	✓	-	**65.3**	**62.3**	**55.7**	**54.5**	**54.0**	**53.9**	**52.1**	**50.6**

(a) Feature distribution trained by cross-entropy loss.

(b) Feature distribution trained by angular penalty loss.

(c) Trained by angular penalty loss with projection layers. Class and data augmentation are applied.

Fig. 6: t-SNE visualization of the feature embeddings for the 60 base classes on CIFAR100. Each small colored number represents one feature instance for that class. The bold black number represents the average prototype for the class.

step. After that, when class augmentation is applied, the average accuracy (harmonic accuracy) is increased by 2.8% (3.2%) in the last step. In addition, when balanced data is used for prototype generation, the harmonic accuracy for both cross-entropy and angular penalty loss trained model is increased. This shows that simply utilizing the same amount of data from the base and incremental sessions to generate class prototypes can effectively alleviate the prediction bias due to data imbalance.

Figure 6 shows the t-SNE visualization of the training data feature generated by different training strategies. The model trained via angular penalty loss makes the training data cluster better in the latent feature space than the model trained via cross-entropy loss. Then with the further help of projection layers, class and data augmentation, diverse and transferable features are obtained while different class features are still well separated.

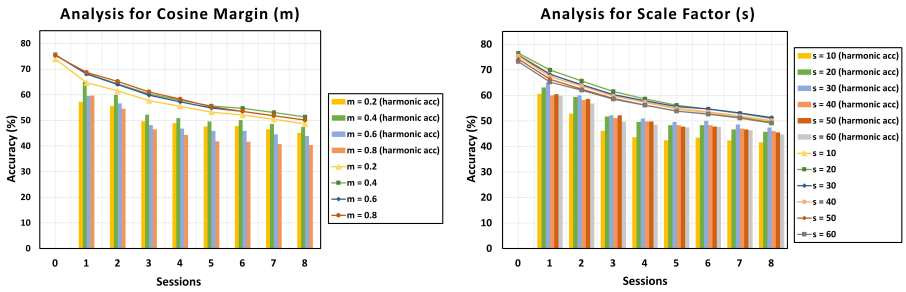

Fig. 7: Hyper-parameter studies for cosine margin (m) and scale factor (s) on CIFAR100 under the 8-step 5-way 5-shot setting.

5.5 Hyper-parameter Analysis

For the angular penalty loss calculation, there are two hyper-parameters — the cosine margin (m) and the scale factor (s). To find the most suitable hyper-parameter value, we perform the hyper-parameter grid analysis on the CIFAR100 dataset under the 8-step 5-way 5-shot protocol. All the experiments for hyper-parameter analysis are trained using angular penalty loss with data augmentation. Figure 7 shows the experimental results. First, we set the scale factor to 30 and vary the value for the cosine margin. According to Figure 7, we find that when the cosine margin is set to 0.4, in most sessions, the best average and harmonic accuracy are acquired. Then, we set the cosine margin to 0.4 and vary the value for the scale factor. We find that when the scale factor is set to 20 or 30, a good performance is usually acquired in most sessions. Thus, for all our experiments, we set the cosine margin to 0.4 and the scale factor to 30.

6 Conclusion

In this paper, we first reformulate the FSCIL task and propose a more practical and comprehensive setup. After that, inspired by techniques from modern face recognition and data augmentation, we proposed our ALICE method. We link the relationship between FSCIL and open-set tasks and emphasize the importance of using base session training to obtain generalizable features for the FSCIL task. We show that with only balanced nearest class mean and no further action in prototype evolution, our method outperforms the SOTA methods by substantial improvements in all benchmark datasets.

Acknowledgments. We thank Dr. Yadan Luo and Kaiyu Guo for their help, discussion, and support. This research was funded by the Australian Government through the Australian Research Council and Sullivan Nicolaides Pathology under Linkage Project LP160101797.

References

1. Castro, F.M., Marín-Jiménez, M.J., Guil, N., Schmid, C., Alahari, K.: End-to-end incremental learning. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 233–248 (2018)
2. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International conference on machine learning. pp. 1597–1607. PMLR (2020)
3. Chen, X., He, K.: Exploring simple siamese representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15750–15758 (2021)
4. Cheraghian, A., Rahman, S., Fang, P., Roy, S.K., Petersson, L., Harandi, M.: Semantic-aware knowledge distillation for few-shot class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2534–2543 (2021)
5. Cheraghian, A., Rahman, S., Ramasinghe, S., Fang, P., Simon, C., Petersson, L., Harandi, M.: Synthesized feature based few-shot class-incremental learning on a mixture of subspaces. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8661–8670 (2021)
6. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4690–4699 (2019)
7. Dong, S., Hong, X., Tao, X., Chang, X., Wei, X., Gong, Y.: Few-shot class-incremental learning via relation knowledge distillation. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 1255–1263 (2021)
8. Goodfellow, I.J., Mirza, M., Xiao, D., Courville, A., Bengio, Y.: An empirical investigation of catastrophic forgetting in gradient-based neural networks. In: Proceedings of International Conference on Learning Representations (2014)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
10. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. NIPS Deep Learning and Representation Learning Workshop (2015)
11. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 831–839 (2019)
12. Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., Kalantidis, Y.: Decoupling representation and classifier for long-tailed recognition. arXiv preprint arXiv:1910.09217 (2019)
13. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
14. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 212–220 (2017)
15. Mai, Z., Li, R., Kim, H., Sanner, S.: Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3589–3599 (2021)
16. McCloskey, M., Cohen, N.J.: Catastrophic interference in connectionist networks: The sequential learning problem. In: Psychology of Learning and Motivation, vol. 24, pp. 109–165. Elsevier (1989)

17. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2001–2010 (2017)
18. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
19. Tao, X., Hong, X., Chang, X., Dong, S., Wei, X., Gong, Y.: Few-shot class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12183–12192 (2020)
20. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011)
21. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5265–5274 (2018)
22. Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., Cheng, Y., Jui, S., Weijer, J.v.d.: Semantic drift compensation for class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6982–6991 (2020)
23. Zhang, C., Song, N., Lin, G., Zheng, Y., Pan, P., Xu, Y.: Few-shot incremental learning with continually evolved classifiers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12455–12464 (2021)
24. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
25. Zhao, H., Fu, Y., Kang, M., Tian, Q., Wu, F., Li, X.: Mgsvf: Multi-grained slow vs. fast framework for few-shot class-incremental learning. arXiv preprint arXiv:2006.15524 (2020)
26. Zhu, F., Cheng, Z., Zhang, X.y., Liu, C.l.: Class-incremental learning via dual augmentation. Advances in Neural Information Processing Systems **34** (2021)
27. Zhu, K., Cao, Y., Zhai, W., Cheng, J., Zha, Z.J.: Self-promoted prototype refinement for few-shot class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6801–6810 (2021)