Progressive Learning Strategy for Few-Shot Class-Incremental Learning

Kai Hu, *Member, IEEE*, Yunjiang Wang, Yuan Zhang, and Xieping Gao, *Member, IEEE*

***Abstract*—The goal of few-shot class incremental learning (FSCIL) is to learn new concepts from a limited number of novel samples while preserving the knowledge of previously learned classes. The mainstream FSCIL framework begins with training in the base session, after which the feature extractor is frozen to accommodate novel classes. We observed that traditional base-session training approaches often lead to overfitting on challenging samples, which can lead to reduced robustness in the decision boundaries and exacerbate the forgetting phenomenon when introducing incremental data. To address this issue, we proposed the progressive learning strategy (PGLS). First, inspired by curriculum learning, we developed a covariance noise perturbation approach based on the statistical information as a difficulty measure for assessing sample robustness. We then reweighted the samples based on their robustness, initially concentrating on enhancing model stability by prioritizing robust samples and subsequently leveraging weakly robust samples to improve generalization. Second, we predefined forward compatibility for various virtual class augmentation models. Within base class training, we employed a curriculum learning strategy that progressively introduced fewer to more virtual classes in order to mitigate any adverse effects on model performance. This strategy enhances the adaptability of base classes to novel ones and alleviates forgetting problems. Finally, extensive experiments conducted on the CUB200, CIFAR100, and miniImageNet datasets demonstrate the significant advantages of our proposed method over state-of-the-art models.**

***Index Terms*—Curriculum learning, few-shot class-incremental learning (FSCIL), robustness, virtual classes.**

I. Introduction

IN RECENT decades, significant progress has been made in the field of deep learning, particularly in various computer vision tasks, such as classification, segmentation, and object detection [1], [2], [3]. The main approach is to iteratively train neural networks on a given dataset until they converge to learn the mapping of large-scale data. Unfortunately, this approach is nonincremental and assumes that all images in all categories can be accessed simultaneously. When faced with a new problem, computers should preferably extract experiences from the past learning similar to humans to rapidly acquire novel knowledge without forgetting previously learned information. However, previous training data are usually inaccessible due to privacy and data security. Consequently, researchers [4] have explored incremental learning as an alternative approach that requires the network to adapt to changes in the distribution of novel data while preventing catastrophic forgetting of previous knowledge. Furthermore, when dealing with rare data (such as military images, medical data, and rare animal photos), there is often not enough novel data available for training. This scarcity increases the likelihood of overfitting incremental data, making such problems more challenging. Researchers refer to this scenario as few-shot class incremental learning (FSCIL).

To address the challenges of forgetting and overfitting in FSCIL, researchers have been working to improve the stability and plasticity of the models. An effective approach is to improve plasticity by fine tuning the model in incremental tasks [5] while constraining the variation of important model parameters through regularization to maintain stability [6], [7]. Another approach is to learn rich, transferable features from the base class [8], [9], freeze the encoding layer, and train the classification layer for incremental tasks. Undoubtedly, the initial knowledge and feature extraction capabilities that the model acquires play a significant role in promoting transfer learning in subsequent tasks. Therefore, when designing the model initialization, the model should effectively capture the features of the base class data, thereby providing a stable and useful foundation of knowledge for future tasks.

Thus, our research focuses on the base sessions. Based on these observations, we found that the previous methods did not take into account the requirements of incremental tasks during the optimization of the base session. This has led to limited optimization directions for incremental tasks. Specifically, these models optimize without considering incremental tasks, solely pursuing optimal performance for the current task, resulting in overfitting and reduced generalization. The overfitting phenomenon of the base data discussed in this article is mainly reflected in the larger feature space occupied by these classes. Further, we found that when these models are

Received 31 July 2024; revised 28 October 2024 and 20 December 2024; accepted 27 December 2024. Date of publication 22 January 2025; date of current version 7 March 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62372170 and Grant 62272404; in part by the Natural Science Foundation of Hunan Province of China under Grant 2023JJ40638; in part by the Major Project of Changsha Science and Technology Bureau under Grant kh2202005; in part by the Research Foundation of Education Department of Hunan Province of China under Grant 23A0146; and in part by the Hunan Province Degree and Postgraduate Teaching Reform Research Project under Grant 2023JGYB132. This article was recommended by Associate Editor P. Shi. *(Corresponding authors: Yuan Zhang; Xieping Gao.)*

Kai Hu, Yunjiang Wang, and Yuan Zhang are with the Key Laboratory of Intelligent Computing and Information Processing of Ministry of Education, Xiangtan University, Xiangtan 411105, China (e-mail: kaihu@xtu.edu.cn; 202121632889@smail.xtu.edu.cn; yuanz@xtu.edu.cn).

Xieping Gao is with the Key Laboratory for Artificial Intelligence and International Communication and the Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, Changsha 410081, China (e-mail: xpgao@hunnu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2025.3525724.

Digital Object Identifier 10.1109/TCYB.2025.3525724

fine tuned with new data, their ability to resist forgetting significantly diminishes. This decline is largely attributed to the increased likelihood of overlap between the base classes and the new classes, leading to greater forgetting. To address this issue, we proposed a two-pronged approach called the progressive learning strategy (PGLS).

First, inspired by the concept of curriculum learning [10], we observed that human learning is a gradual and progressive process. Therefore, we developed a shallow-to-deep learning approach to better retain and understand previously learned knowledge. To simulate the human learning process, our model adopted a progression from simple to complex learning tasks, with the aim of improving overall stability and performance. Specifically, we proposed a covariance noise perturbation method based on statistical information to learn a noise distribution function for each class and perform multiple samplings of the noise function, as illustrated in Fig. 1(a). To assess the robustness of the samples, we retained only perturbations with the highest noise interference. Based on the perturbation results, samples with strong robustness were considered simple, while those with weak robustness were considered challenging. We assigned different weight coefficients to the learning process, as shown in Fig. 1(b). Learning from these simple samples, we can quickly obtain a robust model. Subsequently, we shifted our attention to challenging samples and aim to enhance the generalizability of the model by learning from them. Different from the existing incremental learning methods, our approach effectively mitigated the overfitting problem of the model during the base phase and improves its generalization and stability by mimicking the human learning process.

Second, inspired by the FACT [8], we reserved the feature space for incremental data by introducing virtual classes that achieve forward compatibility. However, our experiments revealed that the introduction of virtual classes had an impact on the learning in real classes, leading to suboptimal performance in the base session. This phenomenon can be attributed to the large number of virtual classes introduced during the early stages of training, resulting in excessive noise, and can lead to suboptimal solutions for the model. Different from the existing methods [8] that utilize a small weight coefficient to suppress the data noise interference, we employed PGLS to gradually introduce virtual classes, thus reducing the noise effect brought by virtual classes. Furthermore, through the gradual introduction of virtual classes, we can simulate the process of introducing incremental data, as shown in Fig. 1(c) and (d). This approach improved the resistance of the model to interference from incremental data and further strengthens its stability.

In summary, the contributions of this study are three-fold as follows.

1) To address the forgetting problem in FSCIL, we proposed a covariance noise perturbation method based on the statistical information to assess the robustness of samples. Using a PGLS that transitions from strong to weak robustness, our method can effectively strengthen the stability of the model.

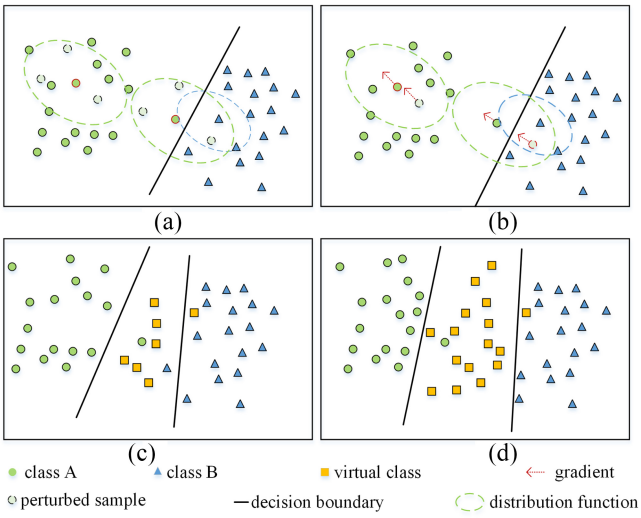

Fig. 1. (a) Distinct noise distribution functions were learned for each category, represented by different colored circles. Multiple perturbation samples were generated by sampling these distribution functions. (b) Using a distance formula, only the perturbations with the highest noise interference were retained. Based on these results, different weights were assigned to the samples. The arrows indicate the gradient's direction and magnitude. (c) Virtual classes for hybrid training are introduced, wherein the reserved space is occupied by generated virtual classes. (d) Our proposed incremental virtual class method gradually squeezes the feature space of real classes, amplifies the differences between classes, and preserves additional feature space.

2) To improve the forward compatibility of the model without compromising its performance, we proposed a progressive virtual incremental method that dynamically extends the reserved space by incrementally increasing the number of virtual classes, which also alleviates the disruption of virtual classes on the model.
3) Extensive experiments conducted on three FSCIL benchmarks, i.e., CUB200, CIFAR100, and miniImageNet, demonstrated that our PGLS outperforms all other approaches and achieves a new state-of-the-art performance.

II. Related Work

In this section, we provided a brief overview of previous research on class-incremental learning (CIL), few-shot learning (FSL), few-shot CIL, and curriculum learning.

A. Class Incremental Learning

As a special form of incremental learning, CIL learns new concepts from nonstatic data streams [11], [12], [13] while retaining the previously learned knowledge of the old classes. Research in this field encompasses a variety of approaches [14], including data-centric, model-centric, and algorithm-centric. Among them, data-centric approaches aim to regulate the model and control the optimization direction using previous data or through data replay and the preservation of knowledge from the previous classes by storing representative samples in a replay buffer. This area of

research includes strategies for selecting appropriate samples or generating new samples, such as the "herding" [15] and "mnemonics" [16] methods, which aim to select representative samples. EASE [13] generated the distribution of old classes in the new classification subspace. The primary goal of these methods is to minimize the forgetting of old tasks while learning new ones. Model-centric approaches [17], [18] allocated distinct sections of the backbone or dynamically expand specific parameters within additional branches for individual tasks. This approach allows each task to be learned independently and reduces the interference between knowledge of different tasks. For example, DER [19] expanded with a new encoder for each task. However, the overhead of such an extended model is substantial. The memory cost of adding another ResNet32 is roughly equivalent to storing 603 CIFAR100 samples [20]. To address this issue, new dynamic expansion methods imposed strict limits on parameter growth. DyTOX [21] expanded by adding a single 384-dimensional task token to each task. In addition, some model-centric methods regularized the specific parameters [6] of the learned task to achieve similar goals. Algorithm-centric methods utilized knowledge distillation to resist forgetting or to rectify biases in the CIL model. LwF [22] used the distillation loss to limit the extent of the model changes, thus mitigating forgetting. UCIR [23] proposed the use of a cosine classifier to avoid the influence of biased classifiers.

B. Few-Shot Learning

The FSL task aims to equip the model with the capability to classify unknown classes using only a limited number of labeled samples. The acquisition of this ability often relies on extensive prior knowledge. Researchers commonly employ a set of base classes with abundant labeled samples to acquire general knowledge. To address the FSL problem, various approaches have been proposed, typically categorized into three main groups.

1) Gradient-based methods [24], [25], [26] prioritized the acquisition of suboptimal parameters customized for individual tasks that serve as the starting point for rapid adaptation to novel tasks. These methods require only a few updates to quickly adapt to the target task. As a prominent gradient-based method, model-agnostic meta-learning (MAML) [24] aims to train a meta-learner that adapts to various learning tasks for fine-tuning updates. To overcome the instability issues associated with MAML, Reptile [27] designed a first-order gradient-based meta-learning method.
2) Metric-based methods [26], [28], [29] seek to achieve this goal by determining an adaptive distance metric across tasks in a discriminative embedding space, thereby achieving outstanding performance in few-shot classification tasks. RelationNet introduced an additional network on top of the representation learning module to learn similarity. The transductive propagation network (TPN) [30] leveraged graphs to propagate labels from labeled to unlabeled samples. Several works aim to enhance class prototype representation for classification by analyzing cosine distances between prototypes. For example, DGIG-Net [31] proposed a new graph prototype framework that embeds visual subgraphs in dynamic metric spaces, replacing traditional linear methods and improving the representativeness and distinguishability of prototypes of the human-object interaction (HOI) class. SCILM [32] generated prototypes by integrating the average mean prototypes with the semantic attention network (SAN)-weighted prototypes, focusing on semantically relevant samples while reducing background noise and outliers for more discriminative visuals. In contrast, our approach enhances representativeness and distinguishability by learning robust samples.
3) Data-augmentation-based methods [33], [34] aim to address the challenge of limited labeled data by generating additional training data or high-quality features. These methods learn a set of effective data augmentation strategies using base-class data and then apply them to few-shot tasks to increase the number of labeled samples. CP-AAN [35] proposed a covariance-preserving adversarial augmentation network to enhance the samples. QFIC [36] replaced traditional data augmentation methods with quantum-enhanced image representations, while MLDSP [37] introduced a distribution distance weighting mechanism that generates source instances more similar to the distribution of the target instance.

C. Few-Shot Class-Incremental Learning

FSCIL [38], [39], [40] aims to address CIL [41] tasks under the condition of insufficient incremental data. Due to the scarcity of training samples, the models tend to overfit incremental samples and forget old knowledge. Current FSCIL methods are mainly divided into two categories: one focuses on fine-tuning the model to adapt to new data while maintaining model stability during incremental sessions, and the other involves training a more plastic backbone network on base classes and then freezing the encoding layers to mitigate the forgetting phenomenon. Furthermore, these methods can be categorized into traditional machine learning methods, meta-learning methods, feature and feature space methods, replay methods, and dynamic network structure methods [42].

The first type of method tends to overfit incremental samples and forget old knowledge owing to the scarcity of incremental training samples. To address this issue, several traditional machine learning methods have been proposed. LPILC [43] proposed the learning of novel classes by adding a new weight column that was solved using linear programming based on existing weights. DMNet [44] decoupled regression and classification tasks, effectively improving the performance of single-stage few-shot object detection. GKEAL [45] treated FSCIL as a recursive learning problem that prevents forgetting. In DBONet [46], an intraclass variance classifier was used to adaptively adjust the class decision boundary and reduce confusion between classes.

Additionally, dynamic network structure methods freeze the previous structure and extend new branch structures to

learn new tasks, thus mitigating disruption to the existing knowledge system. For example, TOPIC [47] used neural gas networks to learn topological knowledge representations of classes formed by different features in the feature space. To address the problem of catastrophic forgetting, these methods stabilize the topological neighborhood graph while adapting it to enhance the discriminative power of the features for a limited number of novel classes. CEC [48] introduced a continuous evolution classifier that adopted a graph model to obtain session classifiers based on pseudo-incremental learning paradigms. Based on the Transformer architecture, CEAT [49] proposed a dynamic expansion network that effectively integrates new branches into the existing structure through reparameterization, maintaining model stability while enhancing its flexibility.

Some researchers address the problem of feature drift, which leads to forgetting from the perspective of features or feature distributions. FSLL [5] suggested selecting a small number of unimportant parameters from the previous tasks to adapt to new tasks, thus alleviating overfitting and forgetting phenomena. WaRP [50] rotated the weight space, compactly pushing the previous knowledge into a space with only a few important parameters. By correctly identifying and freezing these key parameters in the new weight space, WaRP improved the plasticity of the model while ensuring its stability. NC-FSCIL [51] adopted meta-learning methods, preclassifying and fixing the classification space to ensure maximum interclass difference during the training of the encoding layer, thus achieving good predictability and separability.

The second type of method largely addresses the forgetting problem by freezing the encoding layers. F2M [9] employed traditional machine learning methods by introducing noise into the model structure to seek a relatively flat local minimum, allowing model updates within flat regions to reduce knowledge forgetting. MetaFSCIL [52] argues that in incremental learning, the tasks handled during training and testing in the base session are inconsistent, leading to suboptimal solutions. Therefore, they used meta-learning methods to mimic the incremental learning process during the evaluation to ensure consistency between the training and testing tasks. LIMIT [53] also performed a secondary division of the base class data to simulate incremental settings. FSTP-FSCIL [54] and LRT [55] introduced additional models that leverage CLIP to fully exploit the powerful zero-shot capabilities of the large language model, effectively enhancing the model's generalization ability to unseen classes. FACT [8], from the perspective of feature distribution overlap, proposed the concept of forward compatibility by introducing virtual classes during the base session, compressing the feature distribution of the existing classes, and reserving more feature space for incremental sessions.

We also emphasized the base session, which contains abundant samples, to tackle the FSCIL problem. The primary concept is to facilitate the model's step-by-step learning process: first, by categorizing the samples based on robustness and progressively increasing the learning difficulty to enhance model stability. Second, we introduced virtual classes to simulate the incremental process. Unlike LIMIT [53] and MetaFSCIL [52], we did not perform an additional division of the original data. Instead, we aimed for more thorough training during the well-resourced base phase. We incorporated the idea of curriculum learning by dividing the virtual classes based on difficulty and gradually introducing them from easy to hard and from a few to many to expand the reserved space and reduce noise interference in the model. Using this approach, we aim to achieve a more stable model.

D. Curriculum Learning

In this study, we were primarily inspired by the concept of curriculum learning. Curriculum learning, introduced by Bengio [56], mimics the human learning process by allowing models to learn from easy to challenging instances. At a more abstract level, a curriculum can be regarded as a series of instance selections [57] or reweighting strategies to achieve faster convergence and better generalization. The core framework for curriculum learning comprises a difficulty measure and a training scheduler that determines how to structure and learn the curriculum. Curriculum learning is widely used in semi-supervised learning, facilitating the differentiation of confident unlabeled examples and their early or higher-weighted incorporation into the training set while also addressing the issue of noisy pseudo-labels for low-confidence instances by employing harder unlabeled data [58]. Curriculum learning has also found applications in tasks, such as feature selection [59] and domain adaptation [60]. The mechanisms in most studies resemble those of semi-supervised settings, involving the denoising of noisy pseudo-labels [59], [60].

Our PGLS integrated the concept of curriculum learning into FSCIL. We designed covariance-based noise addition as a difficulty measurement tool to assess sample difficulty and reweighting. We then predefined a significant number of virtual classes and employed a linear training scheduler to gradually introduce these virtual classes, progressively increasing their noise level. This approach aimed to systematically enhance the forward compatibility and robustness of the model.

III. Task Description

FSCIL focuses on learning a sequence of disjoint classes with only a few samples available for each class. Specifically, we define a sequence of labeled training datasets as $\{D^0, D^1, \ldots, D^{T-1}, D^T\}$, where $D^t = \{(x_i, y_i)\}_{i=0}^{N_t}$ contains samples from session t. Assuming that the label space of the kth session is denoted by C^k, it is important to highlight that the label sets are disjoint among different sessions, such that $C^i \cap C^j = \emptyset$ $(i \neq j)$.

In the first session D^0, a large-scale training dataset called the base session, is provided. These base classes are used to train the initial model by minimizing the empirical risk of model $f(x)$ on the test set D_t^0 as

$$\min\left(\sum\nolimits_{(x_i,y_i)\in D_t^0} \zeta\left(f(x_i), y_i\right)\right) \tag{1}$$

where $\zeta(\cdot,\cdot)$ measures the difference between the ground-truth and predictions. The model can be decomposed into

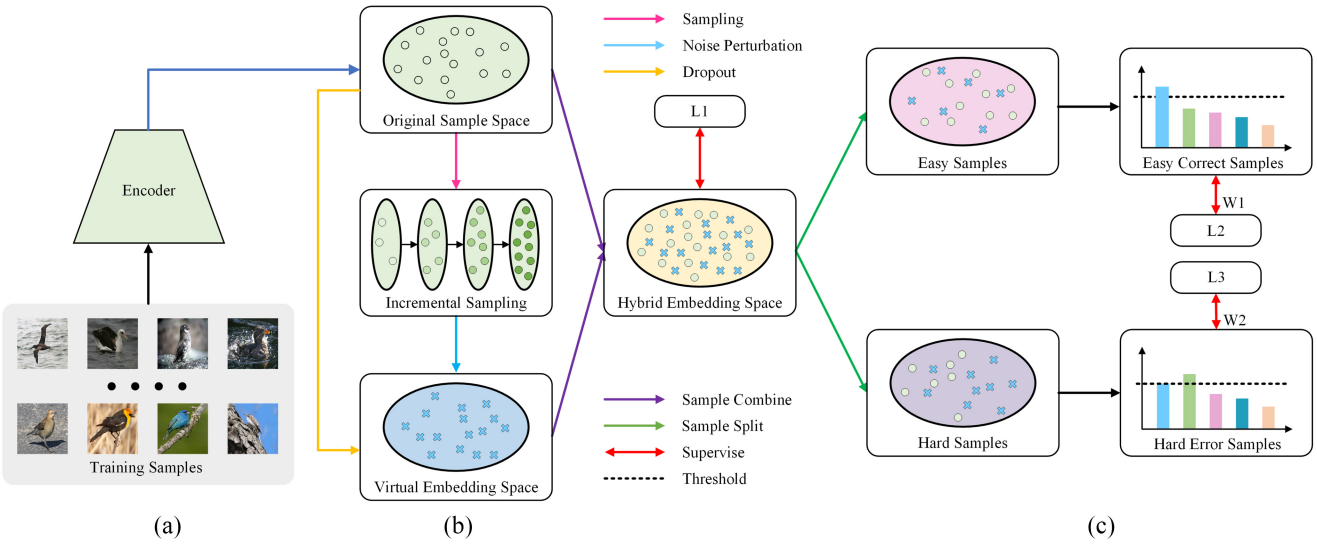

Fig. 2. Overall framework of the proposed progressive learning strategy, which consists of three parts. The first part is the encoding part of the network, which converts images into high-dimensional semantic information. The second part is the virtual class increment section, which gradually generates virtual samples through a progressive sampler. The third part is the robust curriculum learning section, which classifies samples into easy and hard, allowing the model to learn progressively from easier to more challenging tasks.

an embedding layer and a linear classifier, $f(x) = W^T\phi(x)$, where $\phi(\cdot) : \mathbb{R}^D \to \mathbb{R}^d$ and $W \in \mathbb{R}^{d\times|Y_0|}$. The classifier for base session $W_0 = [w_1, \ldots, w_{|Y_0|}]$. Each subsequent session D^t $(t > 0)$ contains only a few novel class training samples with the label C^t. Specifically, the incremental data follow a structured N-way K-shot format, where each session consists of N classes, and within each class, there are K training images. In the mth session, we have access only to D^M. Following learning from D^K, the model undergoes evaluation using test images from all encountered classes up to that point, i.e., $D_t^0 \cup D_t^1 \cup \cdots \cup D_t^M$. The goal of FSCIL is to empower the model to efficiently learn novel classes with limited data in each incremental session while preserving its performance on both the previously learned base and novel classes. The objective is to minimize the empirical risk of overall testing datasets using

$$\min\left(\sum\nolimits_{(x_i,y_i)\in\bigcup_{j=0}^{M} D_t^j} \zeta(f(x_i), y_i)\right). \tag{2}$$

To effectively address the forgetting problem, some researchers have proposed freezing the encoding layers framework, adjusting only the classifier weights during incremental sessions, and extending the classifier in each incremental session: $W = \{W_1^0, W_2^0, \ldots, W_{|C_0|}^0\} \cup \cdots \cup \{W_{|C_{t-1}|+1}^t, \ldots, W_{|C_t|}^t\}$, where the classifier weights are parameterized by the average embedding of each class (i.e., prototype), expressed as

$$W_{|C_k|}^t = \frac{1}{n_{|C_k|}} \sum\nolimits_{i=1}^{n_{|C_k|}} \phi(x_i) \tag{3}$$

where $n_{|C_k|}$ represents the number of training samples in class C_k and $W_{|C_k|}^t$ denotes the classification head for class C_k.

In each session, we used the nearest class mean (NCM) [61] algorithm to assess the accuracy of all classes encountered during the inference process. Specifically, we determine the distance between $\phi(x)$ and all the prototypes presented in the classifiers. To determine the classification result, we find the prototype class that has the highest similarity score with the given input and assign that class as its predicted class using

$$\hat{c} = \arg\max_{i\in 1,\ldots,|C|} d(\phi(x), W_i) \tag{4}$$

where $d(\cdot,\cdot)$ denotes a distance metric, such as the Euclidean distance, and $|C|$ is the total number of classes encountered.

IV. Method

A. Overview

To address the issue of catastrophic forgetting in FSCIL, we proposed a method termed PGLS, which is characterized by two key aspects as follows.

1) *Robust Curriculum Learning:* We adopted the concept of curriculum learning by designing a covariance noise perturbation method to evaluate sample robustness and guide the model to learn step-by-step from easy to hard tasks. Initially, focusing on robust samples helps prevent overfitting of challenging samples and enhances model stability, resulting in more robust decision boundaries.
2) *Incremental Virtual Class:* We incorporated virtual classes for joint training and to mitigate noise interference from the virtual classes, we introduced them in a difficulty-based sequence, from few to many and from easy to hard. This method simulates the continuous introduction of incremental data. Consequently, the forward compatibility and noise resilience of the model are progressively enhanced during the incremental virtual process, leading to better handling of the progressively introduced incremental data.

In contrast to introducing too many virtual classes simultaneously, which may degrade the model's ability to learn from real samples, introducing virtual classes gradually can mitigate this negative effect and enable the virtual classes to better facilitate the forward compatibility of the model. An overview of the proposed PGLS is presented in Fig. 2.

In subsequent incremental sessions, we froze the encoding layers to avoid forgetting due to changes in model parameters. We utilized (3) to obtain prototypes for each incremental class, which provided a more effective starting point for training within incremental learning methods and facilitated faster model convergence. This approach helped alleviate forgetting associated with fluctuations in the model. To demonstrate the effectiveness of our incremental freezing method, we refrained from fine tuning the model during the incremental phase. Consequently, any observed forgetting during training can be attributed solely to the overlap of new and old features. We initialized the classification heads using these prototypes and classified according to (4) to evaluate our method's zero-shot generalization capability.

B. Robust Curriculum Learning

To enhance the model stability and alleviate the forgetting problem in incremental learning, we proposed a PGLS inspired by curriculum learning. Following the learning experience of humans, we progressively guided the model to learn, initially mastering simple knowledge and then gradually progressing to more complex content. This sequential learning helped to strengthen the stability of the model. Our goal was to apply this concept to incremental learning to mitigate forgetting problems. However, the main challenge in curriculum learning is determining the level of difficulty in sample partitioning. Assessing the difficulty level of samples can be challenging, and improper partitioning may result in low learning efficiency and poor performance. Therefore, to achieve the desired outcomes of curriculum learning, a reliable and effective method for dividing the difficulty levels of samples must be developed. To make curriculum learning more suitable for incremental learning requirements, we propose using sample stability to measure the learning difficulty of samples and devise a covariance noise perturbation method based on the statistical information to evaluate sample stability. Covariance is a statistical measure that describes the relationship between two variables. By analyzing the correlation between this variable and the class prototype, we can characterize the data distribution for each class, identify sample difficulty levels, and formulate an effective curriculum learning strategy. The specific implementation is as follows:

$$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \quad (5)$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ represent the mean and covariance of the input data x category, respectively.

First, we used the covariance information from each class to characterize the distribution properties of that class. Here, μ and Σ represent the mean and covariance of different classes, respectively. Specifically, we shift the fluctuations around the mean to the samples, creating covariance noise. The process is as follows:

$$x'_{c_i} = x_{c_i} + \lambda \cdot \mathcal{N}\left(\boldsymbol{\mu}_{c_i}, \boldsymbol{\Sigma}_{c_i}\right).\text{sample}() \quad (6)$$

where $\mathcal{N}(\boldsymbol{\mu}_{c_i}, \boldsymbol{\Sigma}_{c_i}).\text{sample}()$ represents sampling from the distribution function. λ is a scaling factor that we set to one in this context. We added sampled noise to the original data to introduce perturbations. C_i denotes class.

For each x_{c_i}, we performed N samplings, generating N bias vectors to apply to the sample. We supervised the perturbed samples using cosine similarity loss to quantify the degree of noise disturbance. Next, we retained only the sample with the highest noise for subsequent computations. The specific procedure is outlined as follows:

$$X'_{c_i} = \max\left\{\sum_{i=1}^{n}\left(1 - d(x'_{c_i}, \text{center}_{c_i})\right)\right\} \quad (7)$$

where $d(\cdot,\cdot)$ denotes a distance metric that is used to quantify the distance between the perturbed samples and the class mean samples as a measure of sample robustness. Here, we employed the cosine distance. n represents the number of sampling times.

We categorized the perturbed samples into two types based on their classification results after the fully connected (FC) layer and true label results. Subsequently, different weights were assigned to these categories for learning. The specific procedure is as follows:

$$\begin{cases} W1_i = 2, \text{ if } y'_i = y_i \\ W2_i = 1, \text{ if } y'_i \neq y_i \text{ and pred} < 0.5 \end{cases} \quad (8)$$

where y'_i is the predicted result, pred is its confidence, and y_i is its label.

The first category consists of samples whose labels remain correct even after perturbation. It can be reasonably conjectured that these highly robust samples have better decision boundaries than other classes and that focusing on them initially allows the model to have a good initialization, resulting in larger differences between the encoded images for each class. The second category includes samples that have been misclassified following the perturbation and exhibit low confidence levels. A well-designed model has the potential to accurately identify these samples. However, overemphasizing the learning of these samples during the early stages of training can compromise the stability of the model.

We initialized two coefficient matrices $W1$ and $W2$, for the two aforementioned categories, and guided the network for directed learning with different coefficient weights. Specifically, $W1$ and $W2$ were set to 2 and 1, respectively. We encourage the model to focus on easier samples during the initial training phase to facilitate the learning of robust decision boundaries and enhance the stability of the model. For challenging samples, we applied a lower learning rate, ensuring that once they are properly identified as highly robust, the model can further enhance its generalization without compromising stability. The process is as follows:

$$L_{\text{RCL}}\left(X'_i, y_i\right) = (W1 + W2) \cdot \mathcal{L}_{\text{CE}}\left(X'_i, y_i\right) \quad (9)$$

where X'_i represents the sample after covariance perturbation, and y_i is the corresponding true label. $W1$ and $W2$ denote the weight coefficient matrices of different robust samples.

C. Incremental Virtual Class

The introduction of virtual classes can significantly improve the forward compatibility of FSCIL. Inspired by FACT [8],

we preallocated virtual samples within the embedding space, thereby preserving capacity for future classes. Incorporating virtual classes during the early stages of model training allows the model to better adapt to forthcoming incremental learning requirements and mitigates excessive compression of original class embeddings, which reduces the risk that the model forgets prior knowledge. However, it is important to note that adding virtual classes may result in a reduced focus on real classes in the model, which may affect overall performance.

To address this problem, we investigated two approaches for the introduction of virtual classes: 1) coarse-grained and 2) fine-grained. Using the "drop" operation, we obtain a large number of fragments with missing semantic details, which we define as coarse-grained virtual classes. In the early stages of training, we introduced these coarse-grained virtual classes with minimal impact on the model to reduce excessive attention to virtual classes and placeholders. Subsequently, through a linear incremental approach, we gradually introduce virtual classes generated by adding noise that exhibits a high level of realism. Through this process, we progressively introduced fine-grained virtual classes and further refined their occupancies in the feature space. The increment quantity was calculated as $N = \text{Batch} \cdot \max([\text{epoch}/\text{total\_epoch}], 0.2)$ to determine the number of virtual classes introduced incrementally. Throughout the training process, we dynamically increased the value of N to simulate the gradual inclusion of incremental data and the expansion of the reserved space.

Here, are the dynamically growing virtual classes

$$\begin{cases} \hat{x} = \text{concat}((\text{Sample}(x, N) + \text{Noise}), \text{Drop}(x)) \\ \hat{y} = \text{argmax}(\text{concat}(\text{Sample}(x, N), x)[C_{\text{base}}:]) \end{cases} \tag{10}$$

where $\text{Sample}(x, N)$ denotes that in each iteration, we used a sampler to select the top N virtual samples with the highest logit scores (i.e., the N samples with the least interference). We assessed their interference with the model by computing the cross-entropy loss for each virtual class and subsequently sorting them in descending order. In each iteration, we introduced N fine-grained and coarse-grained virtual classes for joint training. Given the characteristics of the various datasets, particularly since all classes represented bird species, which exhibited subtle differences, we set the noise standard deviation to 0.01 for CUB200 and 0.1 for miniImageNet and CIFAR100. As virtual classes are progressively introduced, the model gradually adapts to larger noise levels, thus enhancing its robustness and exhibiting stronger stability in incremental sessions against continuous incremental data. The supervision of the virtual classes is as follows:

$$L_{\text{IVC}}(x, y) = \mathcal{L}_{\text{CE}}(\hat{x}, \hat{y}). \tag{11}$$

D. Algorithmic Flow and Overall Learning Objectives

The training procedure for the base training phase is presented in Algorithm 1. We observed significant differences in distances between classes when projected onto different dataset features. The discriminative criterion for robustness was mainly based on the distance to other classes. When the loss became too large, it further affected the generalization of the model. Thus, we set α to 0.5. Throughout the base training phase, we adhered to the following loss function:

$$L_{\text{total}} = L_{\text{CE}} + \alpha \cdot L_{\text{RCL}} + L_{\text{IVC}}. \tag{12}$$

Algorithm 1 Stage 1 Training Pipeline

Require: μ and Σ_0, where μ is the center and Σ_0 is the covariance function with randomly initialized parameters, D_0, The weight coefficient matrix W initialized with zero weights.

Ensure: encoder $\phi(\cdot)$ and classifier $g(\cdot)$.

for epoch in max epochs **do**
  for $(x, y) \in D_0$ **do**
    $X \leftarrow \phi(x)$
    for i in max sample times **do**
      $X' \leftarrow \max\big(d(X + \text{generate}(\Sigma_0, y), X')\big)$
      $W \leftarrow \text{update}(g(X'), y)$
    end for
    $\hat{X} \leftarrow \text{concat}((\text{Sample}(X, N) + \text{Noise}, \text{Drop}(X)))$
    $\hat{y} \leftarrow \text{argmax}(\text{concat}(\text{Sample}(X, N), X)[C_{\text{base}}:])$
    $L \leftarrow L_{\text{ce}}(g(X), y) + \alpha * W \cdot L_{\text{ce}}(g(X'), y) + L_{\text{ce}}(g(\hat{X}), \hat{y})$
  end for
  $\Sigma_0, \mu \leftarrow \text{update}(\Sigma_0, \mu)$
end for

V. Experiments

In this section, we compared the proposed PGLS with state-of-the-art methods on three benchmark FSCIL datasets. Additionally, we conducted ablation studies to validate the effectiveness of PGLS.

A. Experimental Setup

Datasets: To verify the effectiveness of the proposed method, comprehensive experiments were carried out on three benchmark datasets, namely CUB200-2011 [66], CIFAR100 [67], and miniImageNet [68]. CUB200 is a fine-grained dataset comprising 11 788 images of 200 bird species. It was split into base and incremental sessions with 100 classes each. Each incremental session introduced ten classes, with five training samples and approximately 30 testing samples per class. CIFAR100 and miniImageNet each contained 60 000 images distributed across 100 distinct classes. Sixty classes were allocated to base sessions. Subsequently, for the remaining 40 classes, we conducted incremental training sessions using the five-way five-shot setting, with each class having five training samples and approximately 100 testing samples. The session divisions in all the datasets aligned with the configurations can be found in [47].

Implementation Details: We adopted the ResNet-12 architecture in miniImageNet and CIFAR-100, following the approach outlined in [38], [51]. Additionally, for CUB200, we utilized ResNet-18 (pretrained on ImageNet) based on the prior research. To optimize the model, we utilized stochastic gradient descent (SGD) with a momentum of 0.9. For CIFAR100 and miniImageNet, the initial learning rate was set to 0.1, and cosine annealing was applied to the decay of the

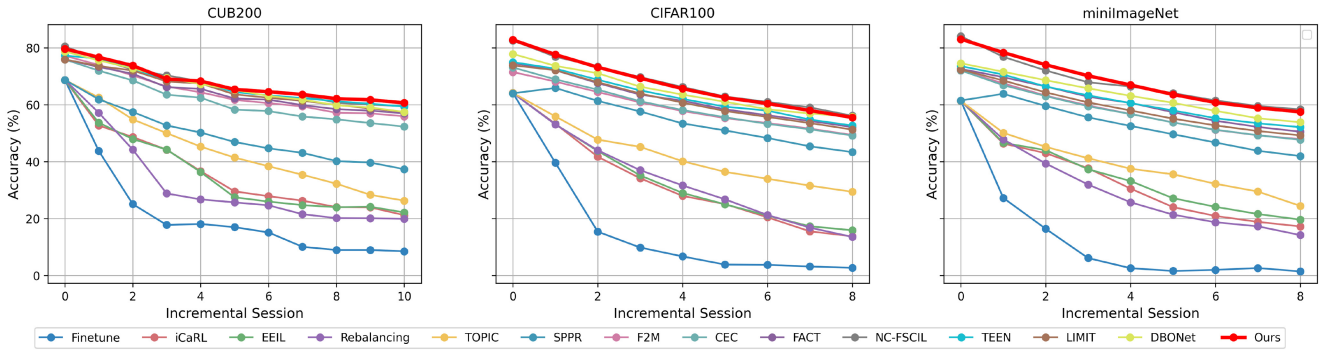

Fig. 3. Comparison with the state-of-the-art methods on the CUB200, CIFAR100, and miniImageNet benchmarks, respectively.

TABLE I
ACCURACY OF THE CUB200 DATASET UNDER THE TEN-WAY FIVE-SHOT INCREMENTAL FEW-SHOT LEARNING SETUP. "PD" REPRESENTS PERFORMANCE DROP AND "AVG" REPRESENTS AVERAGE ACCURACY FROM SESSIONS 0 TO 10

Method	Acc in each session (%)↑											PD↓	Avg↑
	0	1	2	3	4	5	6	7	8	9	10		
Finetune [48]	68.68	43.70	25.05	17.72	18.08	16.95	15.10	10.06	8.93	8.93	8.47	60.21	21.97
iCaRL [63]	68.68	52.65	48.61	44.16	36.62	29.52	27.83	26.26	24.01	23.89	21.16	47.52	36.67
EEIL [64]	68.68	53.63	47.91	44.20	36.30	27.46	25.93	24.70	23.95	24.13	22.11	46.57	36.27
Rebalancing [24]	68.68	57.12	44.21	28.78	26.71	25.66	24.62	21.52	20.12	20.06	19.87	48.81	32.49
TOPIC [48]	68.68	62.49	54.81	49.99	45.25	41.40	38.35	35.36	32.22	28.31	26.26	42.40	43.92
SPPR [65]	68.68	61.85	57.43	52.68	50.19	46.88	44.65	43.07	40.17	39.63	37.33	31.35	49.32
F2M [9]	77.17	73.92	70.27	66.37	64.34	61.69	60.52	59.38	57.15	56.94	55.89	21.24	63.97
CEC [49]	75.85	71.94	68.50	63.50	62.43	58.27	57.73	55.81	54.83	53.52	52.28	23.57	61.33
FACT [8]	75.90	73.23	70.84	66.13	65.56	62.15	61.74	59.83	58.41	57.89	56.94	18.96	64.42
NC-FSCIL [52]	**80.45**	75.98	72.30	**70.28**	68.17	65.16	64.43	63.25	60.66	60.01	59.44	21.01	67.28
TEEN [66]	77.26	76.13	72.81	68.16	67.77	64.40	63.25	62.29	61.19	60.32	59.31	**17.95**	66.62
LIMIT [54]	75.90	73.23	70.84	66.13	65.56	62.15	61.74	59.83	58.41	57.89	56.94	18.96	65.48
DBONet [47]	78.66	75.53	72.72	69.45	67.21	65.15	63.03	61.77	59.77	59.01	57.42	21.24	66.34
PGLS (Ours)	79.59	**76.63**	**73.71**	69.01	**68.28**	**65.34**	**64.52**	**63.57**	**62.10**	**61.69**	**60.62**	18.97	**67.73**

learning rate. For CUB200, we set the initial learning rate to 0.01 and utilized the StepLR learning rate scheduler with a step size of 25.

Evaluation Metrics: Consistent with [47], we denoted the top-1 accuracy after the ith session as $\mathcal{A}_i$. Furthermore, we quantitatively evaluated the forgetting phenomenon using the performance dropping rate (PD), defined as PD $= \mathcal{A}_0 - \mathcal{A}_B$. Here, $\mathcal{A}_0$ represents the accuracy after the base session, and $\mathcal{A}_B$ indicates the accuracy after the final session.

B. Comparison With State-of-the-Art Methods

We compared the proposed PGLS with the existing approaches on three FSCIL benchmarks. These approaches include: classical CIL methods, i.e., iCaRL [62], EEIL [63], Rebalancing [23]; incrementally-trainable FSCIL methods, i.e., TOPIC [47], NC-FSCIL [51], WaRP [50], GKEAL [45], DBONet [46]; and incrementally-frozen FSCIL methods, i.e., SPPR [64], F2M [9], CEC [48], and FACT [8]. Furthermore, we demonstrate a naive baseline method called "Fine tune" [47] that fine tunes with limited data. The performance curves for the three benchmarks are shown in Fig. 3, and the detailed results for the three datasets are presented in Tables I–III, respectively.

Because iCaRL [62], Rebalancing [23], and EEIL [63] were not originally designed to handle FSCIL, their performance was significantly shorter than that of our proposed PGLS. Due to the scarcity of data, the naive fine-tuning approach performed poorly across all results, as the model overfitted the novel classes and quickly forgot the previous ones. Meanwhile, TOPIC [47] focused on forward compatibility while neglecting the importance of backward compatibility, which can lead to overwritten and forgotten previously learned knowledge. DBONet [46] improved on this issue by continuously adjusting the decision boundaries of old classes while learning new ones. However, this approach may compromise the discriminative ability of base classes, essentially sacrificing stability in base classes for the ability to learn new categories. To better accommodate backward compatibility, a series of incrementally-frozen FSCIL methods have been proposed. For example, NC-FSCIL [51] employed high-dimensional orthogonal label vectors to enforce orthogonality between distributions of different classes. This ensures that the base classes are orthogonal to each other, that the base classes are orthogonal to the new classes, and that the new classes are orthogonal to one another. This strategy effectively alleviates the issue of direct overlap between different categories. However, this approach may hinder the learning ability of the model, as the relationships between classes cannot be fully orthogonal. Similarly, our approach adopted an enhanced the discriminative ability of features by precompressing the feature distribution of the base classes. This led to final accuracies of 55.54%, 60.62%, and 58.57% for CIFAR100, CUB200,

TABLE II
ACCURACY OF THE CIFAR100 DATASET UNDER THE FIVE-WAY FIVE-SHOT INCREMENTAL FEW-SHOT LEARNING SETUP. PD REPRESENTS PERFORMANCE DROP AND AVG REPRESENTS AVERAGE ACCURACY FROM SESSIONS 0 TO 8

Method	Acc in each session (%)↑									PD↓	Avg↑
	0	1	2	3	4	5	6	7	8		
Finetune [48]	64.10	39.61	15.37	9.80	6.67	3.80	3.70	3.14	2.65	61.45	16.54
iCaRL [63]	64.10	53.28	41.69	34.13	27.93	25.06	20.41	15.48	13.73	50.37	32.87
EEIL [64]	64.10	53.11	43.71	35.15	28.96	24.98	21.01	17.26	15.85	48.25	33.79
Rebalancing [24]	64.10	53.05	43.96	36.97	31.61	26.73	21.23	16.78	13.54	50.56	34.22
TOPIC [48]	64.10	55.88	47.70	45.16	40.11	36.38	33.96	31.55	29.37	34.73	42.69
SPPR [65]	63.97	65.86	61.31	57.60	53.39	50.93	48.27	45.36	43.32	20.62	54.45
F2M [9]	71.45	68.10	64.43	60.80	57.76	55.26	53.53	51.57	49.35	22.10	59.14
CEC [49]	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	23.93	59.53
FACT [8]	74.60	72.09	67.56	63.52	61.38	58.36	56.28	54.24	52.10	22.50	62.24
NC-FSCIL [52]	82.52	76.82	**73.34**	**69.68**	**66.19**	**62.85**	**60.96**	**59.02**	**56.11**	26.41	**67.50**
TEEN [66]	74.92	72.65	68.74	65.01	62.01	59.29	57.90	54.76	52.64	22.28	63.10
LIMIT [54]	73.81	72.09	67.87	63.89	60.70	57.77	55.67	53.52	51.23	22.58	61.83
DBONet [47]	77.81	73.62	71.04	66.29	63.52	61.01	58.37	56.89	55.78	**20.03**	64.93
PGLS (Ours)	**82.75**	**77.60**	73.14	69.28	65.61	62.49	60.38	57.93	55.54	27.21	67.19

TABLE III
ACCURACY OF THE MINIIMAGENET DATASET UNDER THE FIVE-WAY FIVE-SHOT INCREMENTAL FEW-SHOT LEARNING SETUP. PD REPRESENTS PERFORMANCE DROP AND AVG REPRESENTS AVERAGE ACCURACY FROM SESSIONS 0 TO 8

Method	Acc in each session (%)↑									PD↓	Avg↑
	0	1	2	3	4	5	6	7	8		
Finetune [48]	61.31	27.22	16.37	6.08	2.54	1.56	1.93	2.60	1.40	59.91	13.45
iCaRL [63]	61.31	46.32	42.94	37.63	30.49	24.00	20.89	18.80	17.21	44.10	33.29
EEIL [64]	61.31	46.58	44.00	37.29	33.14	27.12	24.10	21.57	19.58	41.73	34.97
Rebalancing [24]	61.31	47.80	39.31	31.91	25.68	21.35	18.67	17.24	14.17	47.14	30.83
TOPIC [48]	61.31	50.09	45.17	41.16	37.48	35.52	32.19	29.46	24.42	36.89	39.64
SPPR [65]	61.45	63.80	59.53	55.53	52.50	49.60	46.69	43.79	41.92	19.53	52.76
F2M [9]	72.05	67.47	63.16	59.70	56.71	53.77	51.11	49.21	47.87	24.18	57.89
CEC [49]	72.00	66.83	62.97	59.43	56.70	53.73	51.19	49.24	47.63	24.37	57.75
FACT [8]	72.56	69.63	66.38	62.77	60.60	57.33	54.34	52.16	50.49	22.07	60.70
NC-FSCIL [52]	84.02	76.80	72.00	67.83	66.35	64.04	61.46	59.54	58.31	25.71	67.87
TEEN [66]	73.53	70.55	66.37	63.23	60.53	57.95	55.24	53.44	52.08	21.45	61.44
LIMIT [54]	72.32	68.47	64.30	60.78	57.95	55.07	52.70	50.72	49.19	**18.96**	59.06
DBONet [47]	74.53	71.55	68.57	65.72	63.08	60.64	57.83	55.21	53.82	20.71	63.44
PGLS (Ours)	**84.40**	**79.52**	**75.05**	**71.22**	**67.87**	**64.92**	**61.82**	**59.87**	**58.57**	25.83	**69.25**

and miniImageNet, respectively. These results consistently demonstrated performance superior or comparable to those of state-of-the-art methods.

C. Ablation Study

Ablation experiments were conducted to confirm the significance of key components of the proposed method. Within the incremental freezing framework, we used L_{CE} as the baseline for comparison. We found that learning based on L_{CE} did not adequately consider which sample features belonged to data noise, potentially leading to overfitting of hard samples to obtain sensitive decision boundaries.

To address this issue, we introduced the concept of curriculum learning and developed a robust curriculum learning approach by evaluating sample robustness. Furthermore, we formulated the L_{RCL} loss to effectively strengthen the stability of the model. Moreover, to improve the forward compatibility of the model, we adopted a progressive virtual class approach by designing coarse- and fine-grained virtual classes and gradually introducing them. Compared to introducing all virtual classes at once using L_{VC}, L_{IVC} is a fine-tuning method that helps models find optimal solutions within a certain range. Although L_{IVC} may not necessarily outperform L_{VC} when used alone, it is more suitable for combination with other methods without significantly hindering their optimization. The experimental results are presented in Table IV.

To incorporate the concept of curriculum learning into incremental learning, we proposed a covariance noise perturbation method for robustness evaluation. We used the maximum noise interference term as the perturbed sample and employed various distance formulations for robustness scoring. Detailed results are provided in Table V.

We further investigated the effect of robust samples from different partitions on model performance. To this end, we performed ablation experiments on the weight coefficients $W1$ and $W2$ of the different robust samples. We primarily performed two types of ablation comparisons: 1) fixing $W1$ and 2) increasing $W2$, and fixing $W2$ and increasing $W1$. The results have shown that when $W1$ was fixed at 1 and $W2$ was increased, the final results remained relatively stable. In contrast, when $W2$ was fixed and $W1$ increased, the

TABLE IV
ABLATION STUDIES ON THE CUB200 BENCHMARK. IN THE STUDY, VC REFERS TO THE METHOD OF INTRODUCING ALL VIRTUAL CLASSES AT ONCE WITHOUT USING AN INCREMENTAL STRATEGY, IVC REPRESENTS THE INCREMENTAL INTRODUCTION OF VIRTUAL CLASSES, AND RCL STANDS FOR ROBUST CURRICULUM LEARNING

CE	VC	IVC	RCL	Acc in each session (%)↑											PD↓	Avg↑
				0	1	2	3	4	5	6	7	8	9	10		
✓				77.41	74.06	71.04	66.39	66.05	62.95	61.45	60.75	59.08	58.60	57.53	19.88	65.02
✓	✓			78.60	75.59	72.52	67.74	66.51	64.03	63.22	61.56	60.23	59.98	58.83	19.75	66.25
✓		✓		77.95	74.85	71.94	67.52	66.72	63.75	62.68	61.71	60.55	59.76	58.68	19.27	66.01
✓			✓	78.33	75.42	72.22	67.46	66.86	63.71	62.90	61.27	60.85	59.82	58.91	19.42	66.19
✓	✓		✓	79.08	76.48	73.48	**69.15**	67.88	65.32	**64.63**	62.80	61.44	61.47	60.26	**18.82**	67.45
✓		✓	✓	**79.59**	**76.63**	**73.71**	69.01	**68.28**	**65.34**	64.52	**63.57**	**62.10**	**61.69**	**60.62**	18.97	**67.73**

TABLE V
ACCURACY COMPARISON ON THE CUB200 DATASET USING DIFFERENT DISTANCE FORMULAS ACCORDING TO (7)

Method	Acc in each session (%)↑											PD↓	Avg↑
	0	1	2	3	4	5	6	7	8	9	10		
Cosine Similarity	**79.59**	**76.63**	73.71	69.01	68.28	65.34	64.52	63.57	**62.10**	**61.69**	**60.62**	18.97	**67.73**
Manhattan Distance	79.36	76.54	73.58	68.85	68.22	65.03	64.39	63.39	61.59	61.46	60.08	19.28	67.49
Euclidean Distance	79.17	76.57	**73.74**	**69.07**	**68.59**	**65.40**	**64.99**	**63.67**	61.79	61.61	60.36	**18.89**	67.72
Chebyshev Distance	79.00	75.69	72.57	67.64	66.96	63.82	63.20	61.44	60.02	59.59	58.51	20.49	66.22

TABLE VI
ABLATION STUDY OF THE $W1$ AND $W2$ WEIGHT COEFFICIENTS FOR ROBUST SAMPLE DIVISION ON CUB200

$W1$	$W2$	Acc in each session (%)↑											PD↓	Avg↑
		0	1	2	3	4	5	6	7	8	9	10		
0	0	77.41	74.06	71.04	66.39	66.05	62.95	61.45	60.75	59.08	58.60	57.53	19.88	65.02
1	0	79.21	**76.64**	73.60	68.65	68.04	64.81	64.15	62.79	61.35	61.21	60.04	19.17	67.32
1	0.5	78.60	75.84	72.93	68.35	67.54	64.69	64.09	62.43	61.30	61.18	60.17	**18.44**	67.01
1	1	79.14	76.09	72.93	68.28	67.84	64.49	64.25	62.75	61.16	61.14	60.11	19.03	67.11
1	2	79.16	76.32	73.02	68.40	67.65	64.49	63.92	62.59	61.41	61.34	60.27	18.89	67.14
1	5	79.36	76.72	73.53	68.80	68.17	**65.46**	**64.93**	62.99	61.92	**61.69**	**60.64**	18.72	67.65
1	10	78.73	75.92	72.78	68.09	67.11	64.36	63.67	62.37	61.21	60.63	59.38	19.35	66.75
0	1	78.08	75.45	72.48	67.81	66.95	64.32	63.88	62.35	60.81	60.64	59.63	18.45	66.58
0.5	1	78.93	76.31	73.13	68.52	67.92	64.80	64.05	62.69	61.18	61.01	59.99	18.94	67.14
2	1	**79.59**	76.63	**73.71**	**69.01**	**68.28**	65.34	64.52	**63.57**	**62.10**	**61.69**	60.62	18.97	**67.73**
5	1	78.14	74.89	71.32	66.26	65.89	62.57	61.99	61.01	60.04	59.13	57.98	20.16	65.38
10	1	77.52	73.47	69.86	64.90	63.88	60.45	59.50	57.93	56.75	55.88	54.28	23.24	63.13

TABLE VII
ABLATION STUDY RESULTS FOR α SCALING FACTOR ON CUB200

α	Acc in each session (%)↑											PD↓	Avg↑
	0	1	2	3	4	5	6	7	8	9	10		
0	77.95	74.85	71.94	67.52	66.72	63.75	62.68	61.71	60.55	59.76	58.68	19.27	66.01
0.2	78.98	76.31	72.98	68.64	67.94	65.17	**64.71**	62.94	61.88	**61.80**	**60.72**	**18.26**	67.46
0.4	79.19	76.31	73.21	68.31	67.62	64.42	64.06	62.63	61.13	61.42	60.18	19.01	67.13
0.5	**79.59**	**76.63**	**73.71**	**69.01**	**68.28**	**65.34**	64.52	**63.57**	**62.10**	61.69	60.62	18.97	**67.73**
0.6	79.53	76.61	73.43	68.63	67.93	64.64	64.03	62.37	61.15	61.33	60.22	19.31	67.26
0.8	79.26	76.26	72.88	67.90	67.56	64.13	63.37	62.15	60.88	60.76	59.70	19.56	66.80
1.0	79.32	75.83	72.83	68.30	67.28	63.89	63.78	62.67	60.70	60.66	59.30	20.02	66.78
2.0	77.10	73.22	69.88	65.01	63.90	60.38	59.94	58.77	57.41	56.62	55.32	21.78	63.41
5.0	76.80	72.49	68.73	63.58	61.56	57.96	56.94	54.22	53.08	52.05	50.88	25.92	60.75

results initially fluctuated within a small range and then dropped dramatically. The experimental results are presented in Table VI. We speculated that increasing the weight of $W1$ causes the model to focus more on the exclusive features of strongly robust samples, which, to some extent, diminishes the generalization ability of the model and its ability to extract general features. On the other hand, increasing the weight of $W2$ enlarges the area occupied by these samples in the decision space, and when it reaches a certain extent, it is prone to overlapping issues between new and old categories. Therefore, based on the experimental results, we set $W1$ to 2 and $W2$ to 1 as default values for the three datasets.

Additionally, we found that an excessive emphasis on model stability can further compromise model performance. To better balance training stability and model performance, we introduced a scaling factor, denoted α, and performed a series of ablation experiments in α. The results, are shown in Table VII. As α increases, there is a noticeable decline

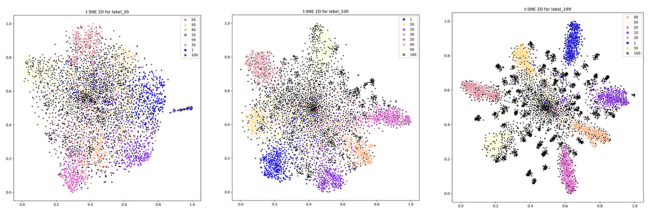

Fig. 4. T-SNE visualization results on the CIFAR-100 dataset, where black points represent virtual classes and other colors denote seven randomly selected real classes. From left to right, the stages are early training, mid-training, and late training.

in overall performance. This decline can be mainly attributed to an excessive emphasis on the robustness of the model, which undermines its generalizability and leads to decreased performance. Our findings suggest that the optimal value for α is 0.5.

Finally, we conducted several experiments to examine the influence of hyperparameters on the proposed method. These results further corroborate the effectiveness of our approach, which is explained in detail in the further analysis section.

D. Visualization

Visualization of Class Separation: To demonstrate the effectiveness of our progressive virtual class approach, we used the CIFAR-100 dataset, which has a wide variety of samples, and performed t-SNE visualizations at the early (20 epochs), mid (100 epochs), and late (199 epochs) stages of training, as shown in Fig. 4.

In the early stages of training, real and virtual classes were not well distinguished. To prevent the model from excessively focusing on virtual classes, we introduced virtual classes with minimal cross-entropy loss. In the mid-training phase, we added virtual classes that closely resembled real samples. This helped the model distinguish and identify classes more effectively, thereby enhancing its resistance to interference. In the late stages, the model learned both virtual and real classes well. We then introduced virtual classes that mostly overlapped with real classes to further increase training complexity and robustness. The classification headers are organized according to virtual labels to facilitate incremental sampling, thereby addressing the model's requirements at various stages.

To provide a more compelling demonstration of the benefits that our approach brings to incremental classification tasks, we chose the CUB200 dataset, which exhibits greater interclass similarity. The embedding space of the CUB200 dataset was visualized using t-SNE [69], as shown in Fig. 5. For this purpose, we randomly selected six base classes and five novel classes. The clustering performance of the cross-entropy was somewhat limited, and the classes were relatively dispersed. In contrast, our PGLS resulted in a more cohesive interclass structure. This is attributed to a decreased emphasis on overly challenging samples, coupled with an increased focus on representative common samples. This approach ultimately leads to the development of a more discriminative decision boundary, thereby reducing the probability of confusion.

Confusion Matrix: In Fig. 6, we presented the confusion matrix generated by both L_{CE} and our proposed PGLS on

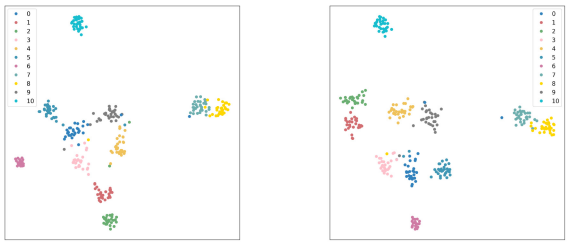

Fig. 5. T-SNE visualization on the CUB200 dataset of the embeddings learned by the CE loss and the proposed PGLS. Classes 0–5 denote the base classes while classes 6–10 represent the novel classes. Our PGLS achieves better class separation. (a) CE. (b) PGLS.

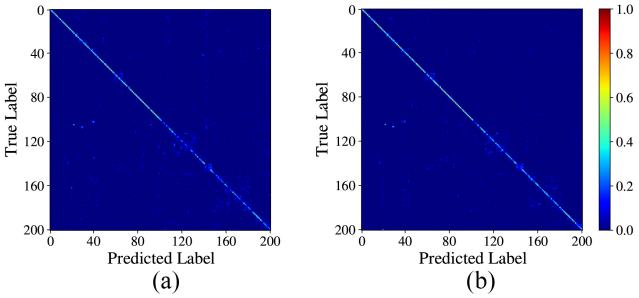

Fig. 6. Comparison of confusion matrices for the CE loss and the proposed PGLS on the CUB200 dataset. Our PGLS exhibits brighter diagonals both in the base categories and the novel categories. (a) CE. (b) PGLS.

the CUB200 dataset after the final training session. The visualization highlights a bright diagonal against a dim background, indicating a higher classification accuracy. In particular, although L_{CE} exhibited a distinct diagonal, its performance in the novel classes was subpar. In contrast, our method showed superior performance across both base and novel classes, showcasing its ability to adapt to novel classes without disrupting previously established decision boundaries.

Performance Measure: The accuracy metric encompasses both learned and novel-added classes. To examine the ability to learn novel classes and resist forgetting, we reported the accuracy of the base and novel classes along with their harmonic means, on the CUB200 dataset. The other settings remained consistent with those of the baseline experiments. The results are shown in Fig. 7. We also presented the results for two competing incremental-freezing based methods: 1) FACT [8] and 2) CEC [48]. From the results, we observed that PGLS outperformed FACT and CEC in both the early and late incremental sessions (Sessions 1 and 10). Moreover, as incremental data were introduced (from Sessions 1 to 10), our method showed greater potential to mitigate base-class forgetting and improve novel class plasticity.

E. Further Analysis

Hyperparameters: Our PGLS has two hyperparameters. The first parameter was utilized to regulate the standard deviation of the Gaussian noise intensity. We adjusted the standard deviation to create differences between the virtual and real classes, and different datasets required varying noise amplitudes to

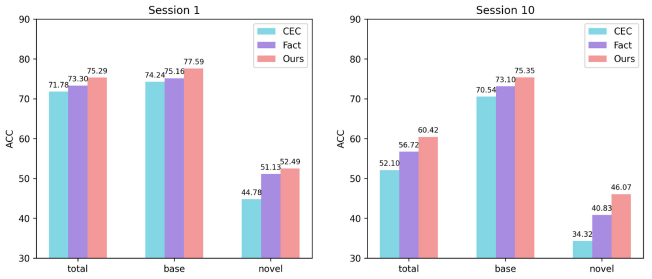

Fig. 7. Illustration of the overall average accuracy, base class average accuracy, and novel class average accuracy for both Sessions 1 and 10 of CUB200.

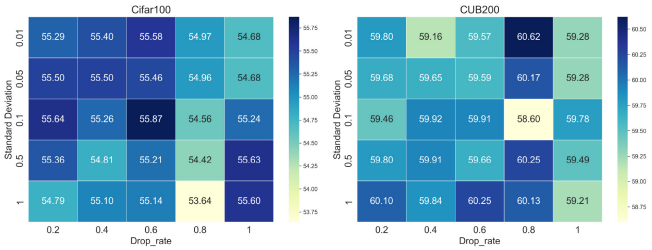

Fig. 8. Ablation experiments of the hyperparameter dropout ratio and hyperparameter noise standard deviation on CIFAR100 and CUB200.

better model the incremental data, generating coarse-grained virtual classes. The second hyperparameter was the dropout ratio. By randomly zeroing a certain proportion of features through dropout, we effectively blurred the details of the real images and attempted to introduce blurred feature information as a coarse-grained virtual class. These coarse-grained virtual classes facilitate rapid generalization and identification of new categories by the model.

We performed ablation experiments on the CIFAR100 and CUB200 datasets to validate our approach by changing the dropout ratio from {0.2, 0.4, 0.6, 0.8, 1} and adjusting the noise standard deviation from {0.01, 0.05, 0.1, 0.5, 1}. We employed dropout to approximate the distribution of virtual classes. A dropout rate that is too high can reduce the fidelity of the data, making the generated data dissimilar to the incremental data. Conversely, a dropout rate that is too low may cause a high similarity between the new virtual classes and the original classes, leading to confusion. Due to the uncertainty of the incremental classes, accurately estimating the distribution relationship between true and incremental categories is challenging. Therefore, we recommend testing different dropout rates for various datasets, with an empirical dropout rate of 0.5 being more suitable. We further enhanced the simulated distribution of these virtual classes by adding noise and tentatively expanding their boundaries using random noise. The experimental results are presented in Fig. 8. We only took the accuracy of the final stage of the increment as the result comparison.

VI. Conclusion

In this study, we successfully incorporated the concept of curriculum learning into incremental learning and proposed a PGLS framework to address the FSCIL problem. Initially, we introduced a covariance noise perturbation method based on the statistical information as a difficulty measurement strategy by applying different perturbations to samples from different classes at different training epochs. We then evaluated the stability of the samples based on their resistance to perturbations, gradually shifting our attention from easy to challenging samples. Consequently, the proposed model is capable of mitigating overfitting associated with challenging samples while establishing a smoother and more discriminative decision boundary. This effectively addresses issues related to sample drift and boundary-crossing, while also alleviating forgetting problems that arise from crossing decision boundaries.

Furthermore, we introduced both coarse-grained and fine-grained virtual classes. The coarse-grained virtual class enhances the model's adaptability, while the fine-grained virtual class is gradually introduced to mitigate the model's excessive reliance on virtual samples, all while preserving the feature space for incremental data embedding. Optimizing the model with this progressive learning strategy helps to find robust local optima, which are crucial to alleviate the forgetting problem. We found that PGLS achieved excellent results on most datasets, but its performance on CIFAR100 was not as good as on the other two datasets. We speculate that this could be due to the significantly lower resolution of the CIFAR100 dataset compared to the others. The reduced resolution leads to loss of finer details and may hinder robust partitioning of samples and generation of more distinct virtual classes. In the future, our efforts will focus on exploring more advanced sample-robust difficulty metrics to enhance the performance of curriculum learning in incremental learning scenarios. In addition, our goal is to design a training scheduling strategy based on the data feedback to dynamically introduce virtual courses to optimize the process more effectively.

Acknowledgment

The authors would like to thank the anonymous reviewers for their insightful comments, which have helped to improve the quality of this article.

References

[1] W. Zhuge, T. Luo, R. Fan, H. Tao, C. Hou, and D. Yi, "Absent multiview semisupervised classification," *IEEE Trans. Cybern.*, vol. 54, no. 3, pp. 1708–1721, Mar. 2024.

[2] B. Xiao, J. Hu, W. Li, C.-M. Pun, and X. Bi, "CTNet: Contrastive transformer network for polyp segmentation," *IEEE Trans. Cybern.*, vol. 54, no. 9, pp. 5040–5053, Sep. 2024.

[3] F. Zuo, J. Liu, Z. Chen, H. Zhang, M. Fu, and L. Wang, "Multilevel fine-grained features-based general framework for object detection," *IEEE Trans. Cybern.*, vol. 54, no. 11, pp. 6921–6933, Nov. 2024.

[4] G. Rypeść, S. Cygert, V. Khan, T. Trzciński, B. Zieliński, and B. Twardowski, "Divide and not forget: Ensemble of selectively trained experts in continual learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2024, pp. 1–18.

[5] P. Mazumder, P. Singh, and P. Rai, "Few-shot lifelong learning," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 2337–2345.

[6] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, 2017.

[7] K. Wei, C. Deng, X. Yang, and D. Tao, "Incremental zero-shot learning," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13788–13799, Dec. 2022.

[8] D.-W. Zhou, F.-Y. Wang, H.-J. Ye, L. Ma, S. Pu, and D.-C. Zhan, "Forward compatible few-shot class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 9046–9056.

[9] G. Shi, J. Chen, W. Zhang, L.-M. Zhan, and X.-M. Wu, "Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima," in *Proc. 35th Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 34, 2021, pp. 6747–6761.

[10] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 41–48.

[11] Z. Mai, R. Li, J. Jeong, D. Quispe, H. Kim, and S. Sanner, "Online continual learning in image classification: An empirical survey," *Neurocomputing*, vol. 469, pp. 28–51, Jan. 2022.

[12] H. Qu, H. Rahmani, L. Xu, B. Williams, and J. Liu, "Recent advances of continual learning in computer vision: An overview," 2021, *arXiv:2109.11369*.

[13] D.-W. Zhou, H.-L. Sun, H.-J. Ye, and D.-C. Zhan, "Expandable subspace ensemble for pretrained model-based class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 23554–23564.

[14] D.-W. Zhou, Q.-W. Wang, Z.-H. Qi, H.-J. Ye, D.-C. Zhan, and Z. Liu, "Deep class-incremental learning: A survey," 2023, *arXiv:2302.03648*.

[15] M. Welling, "Herding dynamical weights to learn," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 1121–1128.

[16] Y. Liu, Y. Su, A.-A. Liu, B. Schiele, and Q. Sun, "Mnemonics training: Multiclass incremental learning without forgetting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 12245–12254.

[17] Z. Wang, T. Jian, K. Chowdhury, Y. Wang, J. Dy, and S. Ioannidis, "Learn-prune-share for lifelong learning," in *Proc. IEEE Int. Conf. Data Min. (ICDM)*, 2020, pp. 641–650.

[18] J. Yu et al., "Boosting continual learning of vision-language models via mixture-of-experts adapters," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2024, pp. 23219–23230.

[19] S. Yan, J. Xie, and X. He, "DER: Dynamically expandable representation for class incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 3014–3023.

[20] D.-W. Zhou, Q.-W. Wang, H.-J. Ye, and D.-C. Zhan, "A model or 603 exemplars: Toward memory-efficient class-incremental learning," 2022, *arXiv:2205.13218*.

[21] A. Douillard, A. Ramé, G. Couairon, and M. Cord, "DyTox: Transformers for continual learning with dynamic token expansion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 9285–9295.

[22] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[23] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 831–839.

[24] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 1126–1135.

[25] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-SGD: Learning to learn quickly for few-shot learning," 2017, *arXiv:1707.09835*.

[26] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017, pp. 1–11.

[27] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," 2018, *arXiv:1803.02999*.

[28] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 1199–1208.

[29] E. Triantafillou, R. Zemel, and R. Urtasun, "Few-shot learning through an information retrieval lens," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 1–11.

[30] J. Kim, T. Kim, S. Kim, and C. D. Yoo, "Edge-labeling graph neural network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 11–20.

[31] X. Liu, Z. Ji, Y. Pang, J. Han, and X. Li, "DGIG-Net: Dynamic graph-in-graph networks for few-shot human–object interaction," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 7852–7864, Aug. 2022.

[32] Z. Ji, X. Yu, Y. Yu, Y. Pang, and Z. Zhang, "Semantic-guided class-imbalance learning model for zero-shot image classification," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 6543–6554, Jul. 2022.

[33] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 7278–7286.

[34] K. Li, Y. Zhang, K. Li, and Y. Fu, "Adversarial feature hallucination networks for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 13470–13479.

[35] H. Gao, Z. Shou, A. Zareian, H. Zhang, and S.-F. Chang, "Low-shot learning via covariance-preserving adversarial augmentation networks," in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NeurIPS)*, vol. 31, 2018, pp. 1–11.

[36] Z. Huang, J. Shi, and X. Li, "Quantum few-shot image classification," *IEEE Trans. Cybern.*, vol. 55, no. 1, pp. 194–206, Jan. 2025.

[37] C. Ren, B. Jiang, N. Lu, S. Simani, and F. Gao, "Meta-learning with distributional similarity preference for few-shot fault diagnosis under varying working conditions," *IEEE Trans. Cybern.*, vol. 54, no. 5, pp. 2746–2756, May 2024.

[38] M. Hersche, G. Karunaratne, G. Cherubini, L. Benini, A. Sebastian, and A. Rahimi, "Constrained few-shot class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 9057–9067.

[39] A. Cheraghian et al., "Synthesized feature based few-shot class-incremental learning on a mixture of subspaces," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (CVPR)*, 2021, pp. 8661–8670.

[40] K. Chen and C.-G. Lee, "Incremental few-shot learning via vector quantization in deep embedded space," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–16.

[41] T. Li, Q. Ke, H. Rahmani, R. E. Ho, H. Ding, and J. Liu, "Else-Net: Elastic semantic network for continual action recognition from skeleton data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (CVPR)*, 2021, pp. 13434–13443.

[42] S. Tian, L. Li, W. Li, H. Ran, X. Ning, and P. Tiwari, "A survey on few-shot class-incremental learning," *Neural Netw.*, vol. 169, pp. 307–324, Jan. 2024.

[43] J. Bai et al., "Class incremental learning with few-shots based on linear programming for hyperspectral image classification," *IEEE Trans. Cybern.*, vol. 52, no. 6, pp. 5474–5485, Jun. 2022.

[44] Y. Lu, X. Chen, Z. Wu, and J. Yu, "Decoupled metric network for single-stage few-shot object detection," *IEEE Trans. Cybern.*, vol. 53, no. 1, pp. 514–525, Jan. 2023.

[45] H. Zhuang, Z. Weng, R. He, Z. Lin, and Z. Zeng, "GKEAL: Gaussian kernel embedded analytic learning for few-shot class incremental task," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2023, pp. 7746–7755.

[46] C. Guo et al., "Decision boundary optimization for few-shot class-incremental learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2023, pp. 3501–3511.

[47] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong, "Few-shot class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020, pp. 12183–12192.

[48] C. Zhang, N. Song, G. Lin, Y. Zheng, P. Pan, and Y. Xu, "Few-shot incremental learning with continually evolved classifiers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 12455–12464.

[49] S. Dong, X. Gao, Y. He, Z. Zhou, A. C. Kot, and Y. Gong, "CEAT: Continual expansion and absorption transformer for nonexemplar class-incremental learning," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Nov. 20, 2024, doi: 10.1109/TCSVT.2024.3502837.

[50] D.-Y. Kim, D. J. Han, J. Seo, and J. Moon, "Warping the space: Weight space rotation for class-incremental few-shot learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023, pp. 1–19.

[51] Y. Yang, H. Yuan, X. Li, Z. Lin, P. Torr, and D. Tao, "Neural collapse inspired feature-classifier alignment for few-shot class incremental learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2023, pp. 1–18.

[52] Z. Chi, L. Gu, H. Liu, Y. Wang, Y. Yu, and J. Tang, "MetaFSCIL: A meta-learning approach for few-shot class incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022, pp. 14166–14175.

[53] D.-W. Zhou, H.-J. Ye, L. Ma, D. Xie, S. Pu, and D.-C. Zhan, "Few-shot class-incremental learning by sampling multiphase tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 11, pp. 12816–12831, Nov. 2022.

[54] H. Ran et al., "Brain-inspired fast-and slow-update prompt tuning for few-shot class-incremental learning," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Sep. 18, 2024, doi: 10.1109/TNNLS.2024.3454237.

[55] Y. Zhao, J. Li, Z. Song, and Y. Tian, "Language-inspired relation transfer for few-shot class-incremental learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 47, no. 2, pp. 1089–1102, Feb. 2025.

[56] G. Alain, A. Lamb, C. Sankar, A. Courville, and Y. Bengio, "Variance reduction in SGD by distributed importance sampling," 2015, *arXiv:1511.06481*.

[57] J. A. Olvera-López, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, and J. Kittler, "A review of instance selection methods," *Artif. Intell. Rev.*, vol. 34, pp. 133–143, Aug. 2010.

[58] C. Gong, D. Tao, S. J. Maybank, W. Liu, G. Kang, and J. Yang, "Multimodal curriculum learning for semi-supervised image classification," *IEEE Trans. Image Process.*, vol. 25, pp. 3249–3260, 2016.
[59] W. Zheng, X. Zhu, G. Wen, Y. Zhu, H. Yu, and J. Gan, "Unsupervised feature selection by self-paced learning regularization," *Pattern Recognit. Lett.*, vol. 132, pp. 4–11, Apr. 2020.
[60] J. Choi, M. Jeong, T. Kim, and C. Kim, "Pseudo-labeling curriculum for unsupervised domain adaptation," 2019, *arXiv:1908.00262*.
[61] S. Guerriero, B. Caputo, and T. Mensink, "DeepNCM: Deep nearest class mean classifiers," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–4.
[62] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 2001–2010.
[63] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 233–248.
[64] K. Zhu, Y. Cao, W. Zhai, J. Cheng, and Z.-J. Zha, "Self-promoted prototype refinement for few-shot class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 6801–6810.
[65] Q.-W. Wang, D.-W. Zhou, Y.-K. Zhang, D.-C. Zhan, and H.-J. Ye, "Few-shot class-incremental learning via training-free prototype calibration," in *Proc. 37th Conf. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–17.
[66] C. Wah, S. Branson, P. Welinder, P. Peronaand, and S. Belongie, "The Caltech-UCSD birds-200-2011 dataset," Dataset, Caltech AUTHORS, Pasadena, CA, USA, 2010.
[67] A. Krizhevsky, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009.
[68] O. Russakovsky et al., "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, pp. 211–252, Dec. 2015.
[69] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

Kai Hu (Member, IEEE) received the B.S. degree in computer science and the Ph.D. degree in computational mathematics from Xiangtan University, Xiangtan, China, in 2007 and 2013, respectively.

He was a Visiting Scholar with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, from 2016 to 2017. He is currently a Professor with the Key Laboratory of Intelligent Computing and Information Processing of Ministry of Education and the School of Computer Science, Xiangtan University. His current research interests include machine learning, pattern recognition, bioinformatics, and medical image processing.

Yunjiang Wang received the B.S. degree in software engineering from Guangdong Neusoft University of Software Engineering, Foshan, China, in 2017. He is currently pursuing the M.S. degree in computer technology with Xiangtan University, Xiangtan, China.

His current research interests include deep learning and medical image processing.

Yuan Zhang received the B.S. degree in biomedical engineering from Zhengzhou University, Zhengzhou, China, in 2009, and the M.S. and Ph.D. degrees in signal and information processing and mathematics from Xiangtan University, Xiangtan, China, in 2012 and 2021, respectively.

She is currently an Assistant Professor with the Key Laboratory of Intelligent Computing and Information Processing of Ministry of Education and the School of Computer Science, Xiangtan University. Her research interests focus on machine learning, bioinformatics, and biomedical image processing.

Xieping Gao (Member, IEEE) was born in 1965. He received the B.S. and M.S. degrees from Xiangtan University, Xiangtan, China, in 1985 and 1988, respectively, and the Ph.D. degree from Hunan University, Changsha, China, in 2003.

He was a Visiting Scholar with the National Key Laboratory of Intelligent Technology and Systems, Tsinghua University, Beijing, China, from 1995 to 1996, and the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, from 2002 to 2003. He is currently a Professor with the Key Laboratory for Artificial Intelligence and International Communication and Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, Changsha. He has authored or co-authored over 150 journal articles, conference papers, and book chapters. His current research interests include areas of wavelet analysis, neural networks, bioinformatics, image processing, and computer networks.

Dr. Gao has been a member of the technical committees of several scientific conferences. He is also a regular reviewer of several journals.