

Feature Drift Oriented Distribution Reconstruction for Imbalanced Class Incremental Learning

Tingmin Li^a, Fengqiang Wan^a, Yipeng Lin^a and Yang Yang^{a,*}

^aNanjing University of Science and Technology

Abstract. Replay-based class incremental learning (CIL) mitigates catastrophic forgetting by retaining limited historical data. However, the class imbalance between historical and new class data prevents catastrophic forgetting from being fully resolved. Existing approaches rely solely on buffer data to approximate the historical distribution, neglecting the incomplete historical distribution modeling caused by feature drift in the evolving feature space, limiting their effectiveness. To address this challenge, we propose a novel Feature Drift oriented Distribution Reconstruction (FDDR) framework, which reconstructs complete historical distributions utilizing both historical and new knowledge. Specifically, FDDR mitigates the structure degradation of historical distributions caused by feature drift by aligning distributions between the current and previous feature spaces, reserving sufficient feature spaces for subsequent distribution reconstruction. Based on the preserved space, FDDR supplements the sparse historical distribution by incorporating weighted new-class distributions according to inter-class relationships, which are derived from joint prototype cosine similarities. In particular, each joint prototype is constructed from both evolved historical prototypes and buffer prototypes, thereby yielding a more reliable similarity measure. Finally, building upon the reconstructed distributions, FDDR generates pseudo-features to correct biased classifier and further fine-tunes the feature space to achieve superior feature space reconstruction. Comprehensive experiments on widely adopted CIL benchmarks verify the effectiveness of FDDR. Code is available at <https://github.com/njustkmg/ECAI25-FDDR>.

1 Introduction

Deep neural networks (DNNs) have recently achieved remarkable success in computer vision tasks under static settings [41, 43, 13, 39]. However, when DNNs are deployed in dynamic and continuously evolving environments, they may fail to continuously learn new knowledge [40, 49]. Class Incremental Learning (CIL) [52, 42, 50] has emerged as a crucial approach for deploying AI systems in dynamic environments, enabling models to acquire new knowledge from a series of tasks. However, training on new tasks will inevitably modify the parameter space of the model, thus compromising previously encoded knowledge, which is a phenomenon known as catastrophic forgetting [38, 32, 29]. Consequently, achieving a stability-plasticity trade-off between preserving historical knowledge and acquiring new knowledge is a fundamental challenge in CIL.

Many techniques have been developed for class incremental learning, including regularization-based methods [21, 2], architecture-

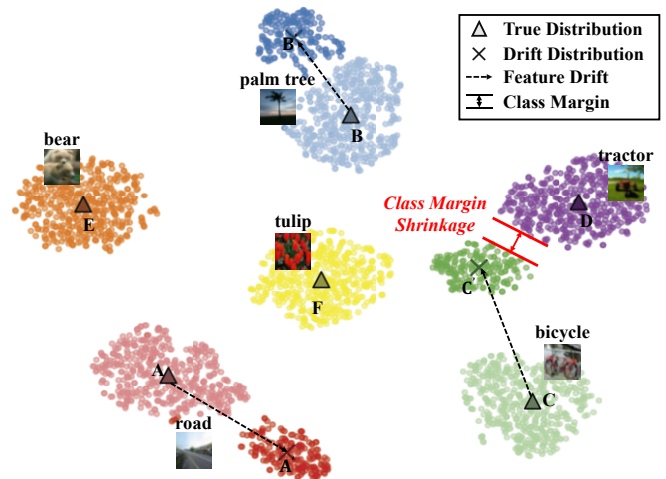


Figure 1. T-SNE visualization of the feature drift phenomenon after learning a new task. A, B, and C represent the feature distributions of the road, palm tree, and bicycle classes following training on Task 1, while A', B', C', D, E, and F represent the distributions of the road, palm tree, bicycle, tractor, bear, and tulip classes following training on Task 2.

based methods [25, 1, 51] and replay-based methods [31, 4, 3]. This paper primarily focuses on replay-based methods, which effectively alleviate catastrophic forgetting by storing a limited number of representative historical samples and replaying them during the learning of new tasks. However, replay-based methods suffer from a severe class imbalance between historical and new class data, which biases the model toward new task, leading to significant performance degradation on historical tasks and weakening the effectiveness of replay-based methods [8, 54].

To address this issue, various approaches have been proposed. Gradient reweighting methods, represented by GR [16] and JIOC [34], aim to achieve balanced optimization by reweighting gradients. Additionally, MAFDRC [7] preserves branch layers for historical tasks to reduce interference between new and historical tasks. However, these methods rely solely on buffer data to model the historical class distribution, overlooking the structure information loss of the historical distribution caused by feature drift, which results in incomplete distribution modeling for historical classes. As illustrated in Figure 1, with the learning of new tasks, feature drift causes the historical distributions to deviate from their original distribution and induces a narrowing of the class margins between historical and new classes, which further exacerbates the difficulty in distinguish-

* Corresponding Author. Email: yyang@njust.edu.cn.

ing historical classes. Therefore, mitigating feature drift problem to reconstruct complete distribution is crucial for achieving balanced learning between historical and new classes.

Targeting this important problem, we propose a Feature Drift oriented Distribution Reconstruction (FDDR) framework, which constructs complete historical distributions by leveraging both historical and new knowledge. Specifically, FDDR leverages the original distribution structure to constrain the historical distribution within the new feature space, thereby mitigating the structure information loss of historical distribution caused by feature drift. Furthermore, FDDR estimates inter-class similarities through joint prototypes, where the incorporation of the evolved historical prototype effectively alleviates similarity bias between classes. By utilizing inter-class similarities, FDDR further supplements the sparse historical distribution with weighted new class distributions, facilitating balanced training.

The contributions of this paper can be summarized as follows:

- We propose a geometric consistency regularization loss that constrains the historical distribution structure to approximate its original form, thereby mitigating the structure information loss of the historical distribution caused by feature drift.
- We design a knowledge transfer-based distribution correction strategy that reconstructs the historical distribution by leveraging similarity-weighted distributions of new classes, where the similarities are computed using joint prototype cosine similarities.
- Extensive experiments on three widely used benchmarks, i.e., CIFAR100, ImageNet100, and ImageNet1000, show the effectiveness of our proposed method, which outperforms the state-of-the-art approaches in imbalanced class incremental learning.

2 Related Work

Class Incremental Learning. Existing CIL methods can be classified into three categories: regularization-based, architecture-based, and replay-based. Regularization-based methods design various strategies to assess important parameters for historical tasks and impose constraints on the updates of these parameters. EWC [21] estimates the importance of parameters using the Fisher information matrix. MAS [2] evaluates the sensitivity of the output function with respect to each parameter. However, such methods often impair the model’s plasticity in adapting to new tasks. Architecture-based methods adjust the model’s representation ability by designing dynamic network architecture. AANets [25] extends the architecture with both stable and plastic blocks, and aggregate their output feature maps to enhance feature representations. Expert Gate [1] employs a set of gating autoencoders to learn task-specific representations and dynamically routes samples to relevant experts during inference. However, these methods introduce additional model parameters and storage overhead, which conflict with the goals of class incremental learning. Replay-based methods approximate historical distributions by storing a limited number of historical samples. GSS [3] introduces a greedy strategy for exemplar sampling. ICaRL [31] selects samples that are closest to the average embeddings of each class. RM [4] enhances the diversity of buffer exemplars by employing perturbation-induced uncertainty and data augmentation strategies. In this paper, we focus on addressing the core issue in the replay-based paradigm, i.e., the class imbalance between historical and new classes, aiming to recover the effectiveness of replay-based methods.

Imbalanced Class Incremental Learning. Learning from imbalanced historical and new data can significantly compromise the effectiveness of replay-based methods. Various methods have been pro-

posed to address this imbalance issue, which can be roughly categorized into four groups: 1) Gradient reweighting. For instance, GR [16] reweights the gradient by using the ratio of the average cumulative gradient between historical and new samples, as well as the ratio of the number of samples; JIOC [34] designs gradient coefficients for different classes based on the gradients of the output scores. 2) Logits adjustment. For example, BDR [53] dynamically adjusts the logits based on the variance of training status and the quantity of classes. FOSTER [33] adds a logits scale factor based on class quantity. 3) Classifier Retraining. MGRB [5] and DER [37] retrain the classifier utilizing balanced training subsets. 4) Parameter Expansion. For example, MAFDRC [7] adds branch layers to prevent interference between new and historical tasks. BiC [36] trains an additional linear model to estimate the classifier bias.

However, these methods approximate the historical distribution solely based on buffer data, thereby implicitly assuming that class imbalance in replay-based methods is a static issue. However, this assumption fails to account for the progressive degradation of learned representations for previously encountered classes over time, induced by feature drift. In contrast, we constrain the structure of the historical distribution by leveraging historical prior knowledge and compensate for its sparsity using knowledge from new classes, aiming to construct a balanced feature space and to mitigate model bias.

3 Method

3.1 Problem Setup and Method Overview

In class incremental learning, the model learns sequentially from a series of T distinct tasks. For each task t , the corresponding dataset is defined as $\mathcal{D}_t = \{(x_i^t, y_i^t), i = 1, \dots, n_t\}$, where x_i^t denotes the i -th image of task t , $y_i^t \in C_t$ represents the corresponding class label, and n_t is the total number of samples in \mathcal{D}_t . The class labels C_t across different tasks are disjoint, i.e., $C_i \cap C_j = \emptyset, \forall i \neq j$. Our approach follows the replay-based class incremental learning paradigm, which maintains a memory buffer \mathcal{M} to store a subset of historical samples, where $|\mathcal{M}| \ll |\mathcal{D}_t|$. The model at task t , denoted as Θ_t , is trained on $\mathcal{D}'_t = \mathcal{D}_t \cup \mathcal{M}$, and consists of a feature extractor F_t and a classification head G_t , i.e., $\Theta_t = F_t \circ G_t$, where \circ denotes function composition. During the evaluation phase, the model is required to classify all previously learned classes $C_{1:t} = \bigcup_{i=1}^t C_i$.

The framework of FDDR is depicted in Figure 2. During the learning of task t , we decouple the feature drift oriented distribution reconstruction process into three stages as follows:

(1) *Feature Drift-Aware Representation Learning Stage.* To mitigate the structure degradation of historical distributions caused by feature drift and in turn reserve feature space for subsequent distribution reconstruction, we design a geometric consistency regularization loss that constrains the structure of historical distribution within the new feature space to align with its original structure. Additionally, we integrate the commonly used model adaptation and fusion (MAF) distillation loss [18, 9] with the cross entropy classification loss in class incremental learning to facilitate the learning of new knowledge while preserving knowledge from historical tasks.

(2) *Classifier Learning Stage.* Guided by the preserved structure of historical distribution, we estimate inter-class similarities based on joint prototype cosine similarity and reconstruct historical distribution by incorporating similarity-weighted new classes distributions. Based on the reconstructed distributions, we further generate pseudo-features for historical classes and fine-tune the classifier.

(3) *Representation Fine-Tuning Stage.* Since only the classifier is fine-tuned during the classifier learning stage, a misalignment arises

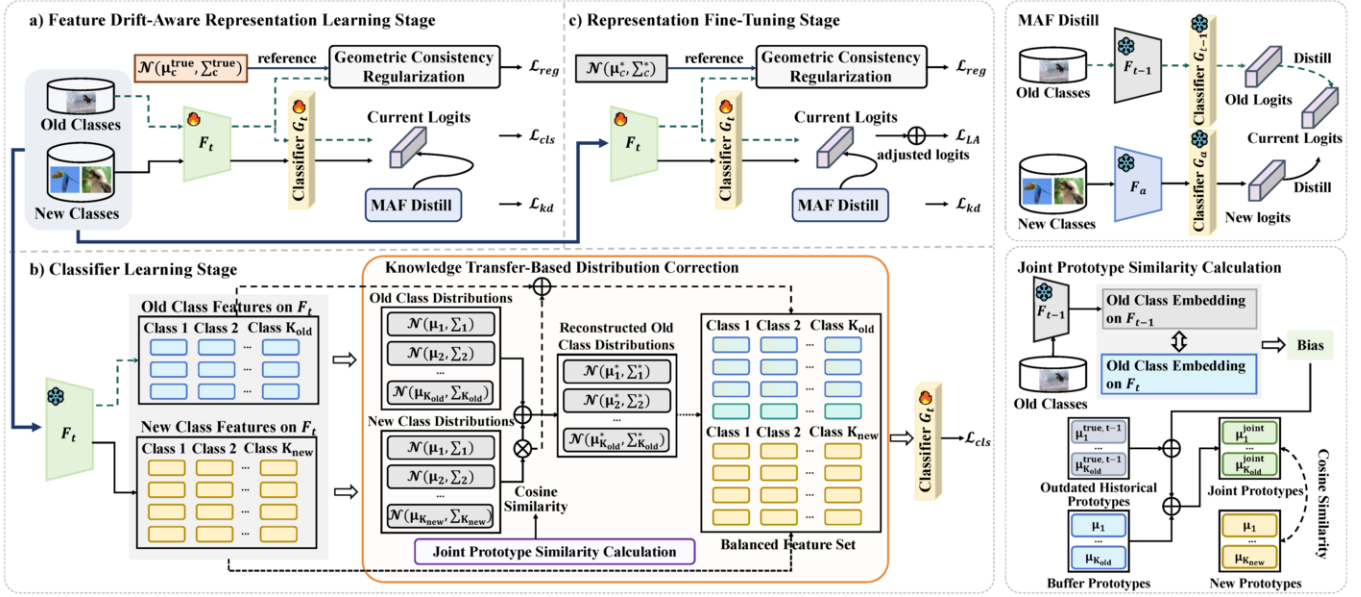


Figure 2. An overview of the FDDR framework. FDDR is achieved through a three-stage training pipeline: (a) In the Feature Drift-Aware Representation Learning Stage, the framework employs geometric consistency regularization loss \mathcal{L}_{reg} to mitigate the distribution information loss caused by feature drift. (b) In the Classifier Learning Stage, the framework utilizes Joint Prototype Similarity Calculation to capture class similarity and applies Knowledge Transfer-based Distribution Correction to supplement the historical distribution with similarity-weighted new class distributions. (c) In the Representation Fine-Tuning Stage, the model is further optimized to align the feature space with the fine-tuned decision boundaries. In this figure, new classes are indexed within the new task, while historical classes are indexed within historical tasks, with K_{new} and K_{old} denoting the number of classes in the new task and historical tasks, respectively.

between the feature extractor and the classifier. Therefore, we further train the model to align the historical distributions with the reconstructed distributions obtained in the second stage, thereby ensuring consistency between the feature space and the decision boundary.

3.2 Feature Drift-Aware Representation Learning

To facilitate the model’s ability to learn new knowledge while preserving historical knowledge, we design a set of optimization objectives. Specifically, our loss function consists of three components: a geometric consistency regularization loss that maintains the structural consistency of historical distributions between the current and previous feature spaces, a cross entropy loss that enables the model to distinguish between new and historical classes, and a MAF distillation loss that distills knowledge from historical and auxiliary models to the new model. The overall objective is as follows:

$$\mathcal{L}_{stage1} = \mathcal{L}_{cls} + \alpha \mathcal{L}_{kd} + \beta \mathcal{L}_{reg}, \quad (1)$$

where α and β are hyperparameters that balance the contributions of the loss functions. Next, we explain the three loss objectives in detail.

Geometric Consistency Regularization Loss. As new tasks arrive, the model parameters are updated to adapt to them. However, this adaptation inevitably causes feature drift in the representations of historical classes [15, 45, 14], which in turn undermines the accurate modeling of historical distributions based on buffer samples. Therefore, we introduce a geometric consistency regularization loss that preserves the alignment of historical distributions between the current and previous feature spaces to alleviate feature drift problem.

After training task $t - 1$, we extract the feature representation of each image x_i^{t-1} using the feature extractor F_{t-1} . Based on these

features, we compute the mean representation μ_c^{true} and the covariance matrix Σ_c^{true} for each class $c \in C_{t-1}$:

$$\mu_c^{true} = \frac{1}{N_c} \sum_{i=1}^{N_c} F_{t-1}(x_i^{t-1}), \quad (2)$$

$$\Sigma_c^{true} = \frac{1}{N_c} \sum_{i=1}^{N_c} (F_{t-1}(x_i^{t-1}) - \mu_c^{true}) (F_{t-1}(x_i^{t-1}) - \mu_c^{true})^T, \quad (3)$$

where N_c denotes the number of samples belonging to class c in the dataset \mathcal{D}_{t-1} . Subsequently, we treat $\mathcal{N}(\mu_c^{true}, \Sigma_c^{true})$ as true prior distribution of the historical class c and store it in the buffer. During the learning of the new task t , to mitigate the structural degradation of historical distributions in the new feature space, we introduce a geometric consistency regularization loss that constrains the historical distribution $\mathcal{N}(\mu_c, \Sigma_c)$ to approximate the true prior distribution:

$$\mathcal{L}_{reg} = \sum_{c=1}^{C_{1:t-1}} (\|\mu_c - \mu_c^{true}\|_2 + \|\Sigma_c - \Sigma_c^{true}\|_F), \quad (4)$$

where ℓ_2 -norm constraint is imposed on the mean representation and Frobenius norm constraint on the covariance matrix. Meanwhile, considering the significant computational overhead associated with computing the distribution statistics of historical classes at each iteration, we employ an online momentum update algorithm to update the distribution $\mathcal{N}(\mu_c, \Sigma_c)$ of historical class batch by batch.

During the learning of batch \mathcal{B} , we compute the local mean representation μ_c^{local} and the local covariance matrix Σ_c^{local} for each historical class c as follows:

$$\mu_c^{local} = \frac{1}{|\mathcal{B}_c|} \sum_{x \in \mathcal{B}_c} F_t(x), \quad (5)$$

$$\Sigma_c^{\text{local}} = \frac{1}{|\mathcal{B}_c|} \sum_{x \in \mathcal{B}_c} (F_t(x) - \mu_c^{\text{local}})(F_t(x) - \mu_c^{\text{local}})^T, \quad (6)$$

where \mathcal{B}_c denotes the subset of samples belonging to class c in the current batch. We subsequently update the global distribution $\mathcal{N}(\mu_c, \Sigma_c)$ of historical classes by integrating the local distribution $\mathcal{N}(\mu_c^{\text{local}}, \Sigma_c^{\text{local}})$, which is formulated as follows:

$$\mu_c \leftarrow w \cdot \text{sg}[\mu_c] + (1 - w) \cdot \mu_c^{\text{local}}, \quad (7)$$

$$\begin{aligned} \Sigma_c \leftarrow w \cdot \text{sg}[\Sigma_c] + (1 - w) \cdot \Sigma_c^{\text{local}} \\ + w(1 - w) \left(\text{sg}[\mu_c] - \mu_c^{\text{local}} \right) \left(\text{sg}[\mu_c] - \mu_c^{\text{local}} \right)^T. \end{aligned} \quad (8)$$

In this equation, $w = N_c^{\text{hist}} / (N_c^{\text{hist}} + |\mathcal{B}_c|)$ denotes the ratio of class c samples from historical batches to all observed batches, where N_c^{hist} and $|\mathcal{B}_c|$ are numbers of class c samples in historical and current batches, respectively, and $\text{sg}[\cdot]$ denotes the stop-gradient operator.

Cross Entropy Loss. Cross entropy loss is commonly employed in classification tasks. Following the replay-based class incremental learning paradigm [7, 20, 35, 11], we optimize the model $\Theta_t = F_t \circ G_t$ on the dataset D'_t by minimizing the cross entropy loss (CE), which is formulated as follows:

$$\mathcal{L}_{\text{cls}} = \sum_{(x,y) \in D'_t} \text{CE}(G_t(F_t(x)), y). \quad (9)$$

MAF Distill Loss. To further preserve the historical knowledge acquired from previous tasks, we adopt the MAF pipeline, which is widely used in class incremental learning [18, 9] and comprises two stages: model adaptation and model fusion. In the model adaptation stage, an auxiliary model Θ_a is introduced and trained exclusively on the new task dataset D_t using the cross entropy loss, enabling it to focus on learning the new task:

$$\mathcal{L}_a = \sum_{(x_i^t, y_i^t) \in D_t} \text{CE}(G_a(F_a(x_i^t)), y_i^t), \quad (10)$$

where F_a and G_a denote the feature extractor and classification head of the auxiliary model Θ_a , respectively. In the model fusion stage, MAF employs knowledge distillation loss to constrain the logits from the new model Θ_t to be consistent with the logits from both the auxiliary model Θ_a and the historical model Θ_{t-1} as follows:

$$\mathcal{L}_{\text{kd}} = \sum_{(x,y) \in D'_t} \left[\text{KL}(p_a^{C_t} \parallel p_t^{C_t}) + \delta \cdot \text{KL}(p_{t-1}^{C_{1:t-1}} \parallel p_t^{C_{1:t-1}}) \right], \quad (11)$$

where δ is a hyperparameter balancing the distill terms, the symbols $p_a^{C_t}$, $p_{t-1}^{C_{1:t-1}}$, $p_t^{C_t}$, and $p_t^{C_{1:t-1}}$ denote the predicted probabilities of auxiliary, historical, and current models for classes C_t and $C_{1:t-1}$, respectively, and KL denotes the Kullback-Leibler divergence.

3.3 Classifier Learning Stage

Previous studies [48, 27, 47] have demonstrated that training on imbalanced datasets tends to skew the classifier's decision boundary toward the new task, leading to performance decline of historical tasks. This issue fundamentally stems from the imbalanced feature distribution between the new and historical classes [6, 30]. To reconstruct a balanced distribution that encourages the classifier to learn all classes equally, we propose a joint prototype similarity calculation strategy and knowledge transfer-based distribution correction strategy, aiming to identify semantically similar new classes and transfer their class distributions to compensate for the incomplete historical distribution, as detailed in the following sections.

3.3.1 Joint Prototype Similarity Calculation

Traditional methods [8, 26] estimate class similarities using the confusion matrix derived from the classification head. However, these similarities are often inaccurate due to inherent biases of the classifier. Decoupled learning [19] strategy demonstrates that models can acquire high-quality feature representations even under data imbalance conditions, which motivates us to estimate joint prototype similarity derived from the feature space to mitigate similarity bias. The joint prototype consists of two components: the prototype computed from buffer data and the evolved historical prototype.

To obtain the prototype μ_c computed from buffer data, we average the embeddings of the corresponding samples belonging to class c , following the formulation in Equation 2 and replacing the dataset with \mathcal{M} . However, as new tasks emerge, the memory allocated to each class progressively decreases, leading to buffer prototypes that fail to accurately represent their corresponding classes. Therefore, we propose incorporating an evolved historical prototype, which is progressively updated based on the outdated true prototype originally derived from the complete class data, in order to enhance the representation capacity of class prototypes.

To obtain the evolved historical prototype $\mu_c^{\text{true},t}$ for task t , we update the outdated historical true prototypes $\mu_c^{\text{true},t-1}$ from the previous task $t-1$ by estimating the feature drift vector between the historical and new feature extractors. The outdated historical true prototypes are initially computed using Equation 2 and their task identity is recorded. In the replay-based class incremental learning paradigm, we have access to authentic historical data in the buffer. Therefore, we estimate the feature drift by utilizing the buffer data as proxy:

$$\Delta\Theta_c^{t-1 \rightarrow t} = \frac{1}{|\mathcal{M}_c|} \sum_{x \in \mathcal{M}_c} (F_t(x) - F_{t-1}(x)), \quad (12)$$

where $\Delta\Theta_c^{t-1 \rightarrow t}$ denotes the feature drift of class c between the historical feature extractor F_{t-1} and new feature extractor F_t , and \mathcal{M}_c represents the set of samples belonging to class c stored in the buffer. The evolved historical prototype $\mu_c^{\text{true},t}$ is updated as follows:

$$\mu_c^{\text{true},t} = \mu_c^{\text{true},t-1} + \Delta\Theta_c^{t-1 \rightarrow t}. \quad (13)$$

Subsequently, for historical class $c \in C_{1:t-1}$, we integrate the evolved historical prototype with the prototype derived from buffer data to construct the joint prototype:

$$\mu_c^{\text{joint}} = \eta \mu_c^{\text{true},t} + (1 - \eta) \mu_c, \quad (14)$$

where $\eta \in [0, 1]$ is a mixing hyperparameter. The similarity between historical class i and new class j is defined as $S_{ij} = \frac{\mu_i^{\text{joint} \top} \mu_j}{\|\mu_i^{\text{joint}}\|_2 \|\mu_j\|_2}$.

3.3.2 Knowledge Transfer-based Distribution Correction

Although the distribution structure is preserved during the feature drift-aware representation learning stage, the historical distribution remains inherently sparse due to the limited amount of data retained in the buffer. Motivated by the fact that new classes exhibit diverse features [44, 24, 23], which are reflected in their covariance matrices, we compute the covariance matrix compensation term for historical class c based on the similarity-weighted covariance matrix of new classes. This term is formulated as $\Delta\Sigma_c = \sum_{j=1}^{C_t} \hat{S}_{cj} \Sigma_j$, where \hat{S}_{cj} represents the normalized similarity between class c and class j .

Meanwhile, considering that historical classes may lack critical features for variation based on the covariance matrix, we also introduce a feature representation compensation term for each historical

class c , computed as $\Delta\mu_c = \sum_{j=1}^{C_t} \hat{S}_{cj}\mu_j$. Subsequently, we obtain the reconstructed historical distribution $\mathcal{N}(\mu_c^*, \Sigma_c^*)$ by incorporating both covariance matrix term and feature representation compensation term, where $\mu_c^* = \mu_c + \gamma\Delta\mu_c$ and $\Sigma_c^* = \lambda(\Sigma_c + \Delta\Sigma_c)$. Here, γ and λ represent the feature representation rectification coefficient and the covariance scaling factor, respectively.

Based on the reconstructed distribution, we further construct a balanced feature set for classifier fine-tuning. First, we extract the original features for both the new classes, denoted as $\mathcal{F}_{\text{new}}^{\text{orig}}$, and the historical classes, denoted as $\mathcal{F}_{\text{old}}^{\text{orig}}$. Next, for each historical class c , we augment its features $\mathcal{F}_{\text{old},c}^{\text{orig}}$ using the feature representation compensation term, which is given by:

$$\mathcal{F}_{\text{old},c}^{\text{aug}} = \mathcal{F}_{\text{old},c}^{\text{orig}} + \gamma\Delta\mu_c, \quad (15)$$

where γ denotes the feature rectification coefficient introduced previously. We then aggregate the augmented features from all historical classes as $\mathcal{F}_{\text{old}}^{\text{aug}} = \bigcup_c \mathcal{F}_{\text{old},c}^{\text{aug}}$. In addition, to balance the number of features between historical and new classes, we sample pseudo-features from the reconstructed distributions for each historical class, i.e., $\mathcal{F}_{\text{old}}^{\text{syn}} \sim \mathcal{N}(\mu_c^*, \Sigma_c^*)$. Finally, the balanced feature set is constructed as $\mathcal{F}_{\text{bal}} = \mathcal{F}_{\text{old}}^{\text{syn}} \cup \mathcal{F}_{\text{old}}^{\text{aug}} \cup \mathcal{F}_{\text{new}}^{\text{orig}}$, which is used to fine-tune the classifier and mitigate classifier bias. The optimization objective of this stage is formulated as follows:

$$\mathcal{L}_{\text{stage2}} = \sum_{(z,y) \in \mathcal{F}_{\text{bal}}} \text{CE}(G_t(z), y), \quad (16)$$

where z denotes the feature and y represents its corresponding label.

3.4 Representation Fine-Tuning Stage

After the classifier training stage, a misalignment issue arises between the feature extractor and the classifier. Therefore, we further fine-tune the model to align the feature representations with the refined decision boundaries by aligning the historical class distributions with the reconstructed distributions as formulated below:

$$\mathcal{L}_{\text{reg}} = \sum_{c=1}^{C_{1:t-1}} (\|\mu_c - \mu_c^*\|_2 + \|\Sigma_c - \Sigma_c^*\|_F). \quad (17)$$

Furthermore, we integrate logits adjustment [28] with cross-entropy loss to mitigate the risk of classifier bias reoccurrence:

$$\mathcal{L}_{\text{LA}} = \sum_{(x,y) \in D'_t} \text{CE}(G_t(F_t(x)) + \tau \cdot \ln \psi_c, y), \quad (18)$$

where $\psi_c = \frac{N'_c}{\sum_{i=1}^{C_{1:t}} N'_i}$ represents the frequency of class c , with N'_i denoting the number of samples of class i in dataset D'_t , and τ is a hyperparameter. Additionally, we also apply the MAF distillation loss \mathcal{L}_{kd} during this phase. The overall objective at this stage can be expressed as:

$$\mathcal{L}_{\text{stage3}} = \mathcal{L}_{\text{LA}} + \alpha\mathcal{L}_{\text{kd}} + \beta\mathcal{L}_{\text{reg}}, \quad (19)$$

where α and β are hyperparameters that retain the same values as those used in the first stage, thus avoiding the introduction of additional hyperparameter and simplifying the optimization process.

4 Experiments

In this section, we first describe our experimental setup, then present the main results on multiple CIL benchmarks, followed by ablation studies and further analysis.

4.1 Experimental Setup

Learning Settings and Datasets. In line with the class incremental learning literature [7], we evaluate our method on three public datasets: CIFAR100 [22], ImageNet100 [10], and ImageNet1000 [10]. Both ImageNet100 and CIFAR100 contain 100 categories, and we adopt two widely used CIL protocols. **B0** (Base 0): In this protocol, the datasets are divided into 5, 10, or 20 tasks, with each task containing 20, 10, or 5 classes, respectively. The memory buffer size is fixed at 2,000 exemplars. **B50** (Base 50): The initial task comprises 50 categories, and the remaining 50 categories are split across 5 or 10 tasks, with each task containing 10 or 5 classes, respectively. The memory size is set to 20 exemplars per class. For ImageNet1000, which consists of 1,000 categories, we follow the B0 protocol and split the dataset into 10 tasks, with each task containing 100 categories. The memory buffer size is fixed at 20,000 exemplars.

Evaluation Metric. To compare with other approaches, we adopt two commonly used evaluation metrics in CIL: Avg accuracy, which is the average accuracy of all tasks, and Last accuracy, which is the final accuracy after the last task. Formally, Avg accuracy and Last accuracy are defined as follow:

$$A_{\text{last}} = \frac{1}{T} \sum_{t=1}^T a_{T,t}, \quad A_{\text{avg}} = \frac{1}{T} \sum_{i=1}^T \left(\frac{1}{i} \sum_{j=1}^i a_{i,j} \right), \quad (20)$$

where $a_{i,j}$ is the test accuracy for task j after training on task i .

Baselines. We compare FDDR with state-of-the-art replay-based methods: ICaRL [31], BiC [36], WA [46], PODNet [12], DER w/o P [37], FOSTER B4 [33], FOSTER [33], and MAFDRC [7]. Specifically, FOSTER B4 refers to the model obtained by FOSTER before feature compression stage, and DER w/o P refers to DER without pruning. For consistency with previous work [7], we use a modified 32-layer ResNet32 [31] as the backbone for the CIFAR dataset, and ResNet18 [17] as the backbone for the ImageNet dataset.

4.2 Main Results

In this section, we report the performance of FDDR on CIFAR benchmarks and the challenging ImageNet benchmark.

For the large-scale ImageNet benchmark, Table 1 presents the performance of FDDR on the ImageNet100 and ImageNet1000 datasets. On the ImageNet100 dataset, FDDR outperforms existing methods by a considerable margin across multiple benchmarks. Specifically, under the B0 5 steps setting, FDDR achieves an improvement of approximately 0.64% in Avg accuracy and 1.35% in Last accuracy compared to the best-performing baseline, MAFDRC. Under the B0 10 steps setting, FDDR raises the Avg accuracy from 79.66% to 80.19%, and the Last accuracy from 70.41% to 72.10%. Moreover, on the ImageNet1000 dataset with B0 10 steps setting, FDDR consistently outperforms the MAFDRC method in terms of both Avg accuracy and Last accuracy. These results clearly demonstrate the state-of-the-art performance of FDDR on large-scale datasets.

For the CIFAR benchmark, Table 2 reports a detailed comparison of performance across various settings. As we can see, FDDR achieved the best or second-best performance across all baselines, excluding DER, which extends the feature extractor and requires approximately k times the parameters of our method for the k -th task, resulting in significantly higher model complexity. Compared to MAFDRC, which is the most competitive method except for DER, FDDR achieves improvements of 0.81% in Last accuracy and 0.43% in Avg accuracy under the B50 5 steps setting. Remarkably, FDDR

Methods	ImageNet100 B0						ImageNet100 B50				ImageNet1000	
	5 steps		10 steps		20 steps		5 steps		10 steps		10 steps	
	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last
ICaRL	74.87	63.36	70.35	55.78	67.80	51.78	64.69	54.46	57.92	50.52	54.15	36.25
BiC	77.11	67.10	70.98	52.00	63.79	41.70	68.51	54.36	60.73	43.04	61.66	41.30
WA	77.59	68.36	73.59	60.78	68.81	57.16	68.49	59.74	62.10	54.42	59.23	40.92
PODNet	76.73	64.90	70.13	53.30	62.78	47.10	78.41	69.18	75.97	66.50	-	-
DER w/o P	81.03	74.44	78.30	70.40	78.22	71.40	80.30	74.28	78.58	<u>71.66</u>	67.41	58.56
FOSTER B4	79.59	72.58	76.54	67.08	74.21	62.16	79.93	72.48	76.27	67.04	68.34	58.53
FOSTER	78.38	71.38	76.22	66.70	73.95	62.42	79.56	71.18	75.79	66.90	-	-
MAFDRC	<u>82.22</u>	<u>76.01</u>	<u>79.66</u>	<u>70.41</u>	75.21	63.59	81.37	<u>74.86</u>	<u>77.95</u>	71.26	<u>69.37</u>	<u>59.59</u>
FDDR	82.86	77.36	80.19	72.10	<u>76.18</u>	<u>64.60</u>	<u>80.37</u>	75.64	77.70	71.72	69.62	59.60

Table 1. Results on the ImageNet100 and the ImageNet1000 datasets. The best results are highlighted in bold, and the second-best results are underlined.

Methods	CIFAR100 B0						CIFAR100 B50			
	5 steps		10 steps		20 steps		5 steps		10 steps	
	Avg	Last	Avg	Last	Avg	Last	Avg	Last	Avg	Last
ICaRL	69.29	57.03	68.37	53.04	67.43	49.65	62.21	53.63	53.65	47.18
BiC	68.66	58.22	67.75	53.31	65.41	47.12	63.92	54.18	59.68	48.04
WA	72.09	61.49	70.88	56.74	68.10	49.60	67.30	59.37	61.86	50.86
PODNet	69.32	57.75	63.17	47.49	58.26	40.62	70.40	62.49	69.20	60.14
DER w/o P	75.83	68.95	75.71	65.85	74.04	62.53	72.95	68.06	72.50	67.37
FOSTER B4	74.53	65.31	73.13	61.81	70.64	56.84	71.31	64.66	68.90	61.41
FOSTER	72.46	63.35	71.80	60.15	69.56	56.50	70.09	63.63	68.05	60.71
MAFDRC	74.87	66.45	73.97	62.04	71.75	<u>57.65</u>	71.65	65.09	<u>70.21</u>	62.20
FDDR	<u>75.39</u>	<u>66.48</u>	<u>74.40</u>	<u>62.72</u>	<u>71.76</u>	57.09	<u>72.08</u>	65.90	69.74	<u>62.55</u>

Table 2. Results on the CIFAR100 B0 and B50 settings. The best results are highlighted in bold, and the second-best results are underlined.

avoids introducing extra branch parameters like MAFDRC, and consistently outperforms the model expansion approach FOSTER B4 across all settings, demonstrating its effectiveness in imbalanced class incremental learning.

4.3 Ablation Study

Each Component of FDDR. We conduct a comprehensive evaluation of each component in our framework, including: (i) the geometric consistency regularization (GCR) loss; (ii) the KTDR-JPS strategy, which integrates knowledge transfer-based distribution correction (KTDR) strategy with joint prototype similarity calculation (JPS) strategy; and (iii) the logit adjustment (LA) loss applied during the representation fine-tuning stage. Specifically, we adopt MAF distill loss combined with cross entropy loss as the baseline and evaluate the effectiveness of each component under the B0 10 steps setting on CIFAR100 and ImageNet100 datasets. The results are presented in Table 3. The performance of our full FDDR framework is significantly better than that of the vanilla backbone (results in the first row). Furthermore, performance improvements are observed for each component. For example, on the CIFAR100 dataset, the Avg accuracy improves from 67.11% to 72.65% with KTDR, is further boosted to 72.97% by leveraging the structural information provided by GCR, and is enhanced by an additional 1.43% with the LA loss.

Sensitivity to Hyper-Parameters γ and λ . Considering that γ and λ determine the compensation scales for the feature representation and covariance matrix, respectively, we evaluate the sensitivity of FDDR to these hyperparameters under the B0 10 steps benchmark on the CIFAR100 dataset. Specifically, we set γ to different values, i.e., $\{0.4, 0.5, 0.6, 0.7, 0.8\}$, and λ to $\{0.8, 0.9, 1.0, 1.1, 1.2\}$. As

Table 3. Ablations of the different components in FDDR under the B0 10 steps setting on the CIFAR100 and ImageNet100 datasets.

KTDR	GCR	LA	CIFAR100		ImageNet100	
			Avg	Last	Avg	Last
			67.11	49.56	67.75	49.60
✓			72.65	59.90	76.63	63.04
✓	✓		72.97	60.09	76.92	63.42
✓	✓	✓	74.40	62.72	80.19	72.10

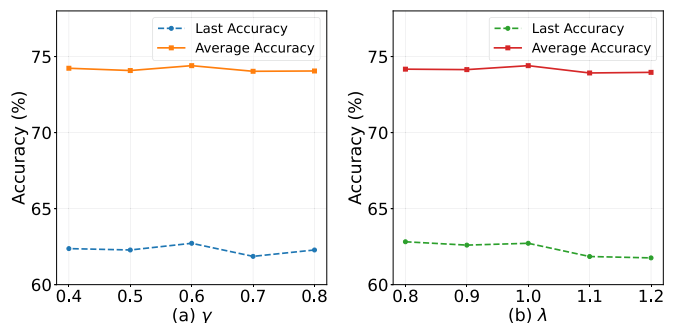


Figure 3. Parameter Sensitivity of FDDR under the B0 10 steps setting on CIFAR100 dataset. (a) reports the Last accuracy and Avg accuracy for different values of γ , and (b) reports the Last accuracy and Avg accuracy for different values of λ .

shown in Figure 3, we observe the following: (1) $\gamma = 0.6$ achieves the best Avg accuracy and Last accuracy, suggesting that appropriately compensating for historical features is beneficial for overall performance. (2) $\lambda = 1$ yields the highest Avg accuracy of 74.40%

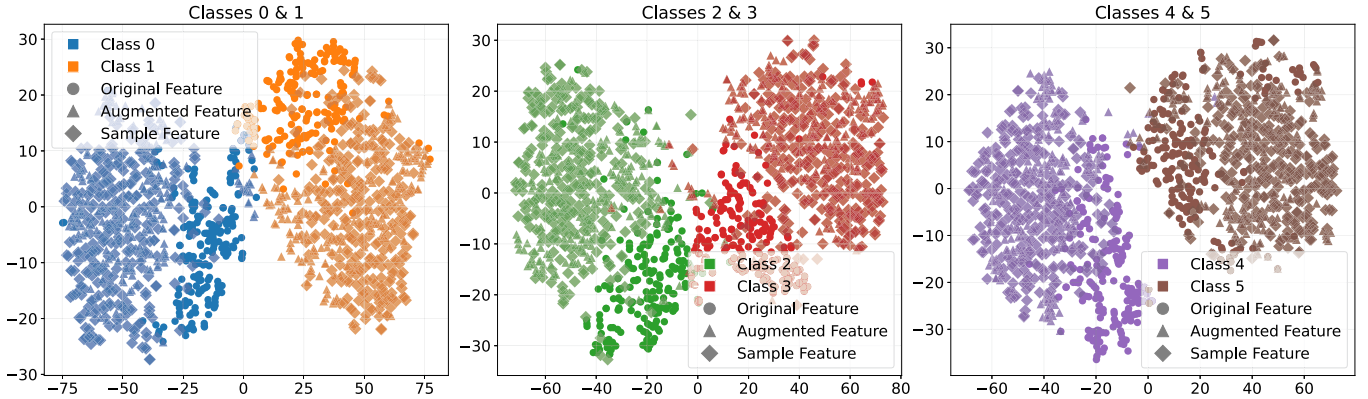


Figure 4. T-SNE visualization of historical distribution reconstruction on CIFAR100 after training on Task 2. Sampled features are denoted by diamonds, augmented features by triangles, and original features by dots.

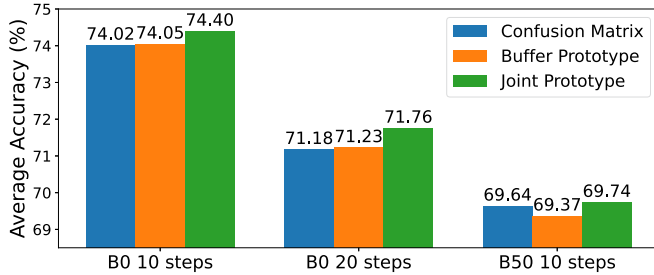


Figure 5. Compare Joint Prototype Cosine Similarity (JPS) with the Confusion Matrix (CM) and the Prototype computed from buffer data (BP). The evaluation criterion is Avg Accuracy.

and the second highest Last accuracy of 62.72%. Moreover, further increasing λ results in a decline in overall performance, indicating that excessive compensation for historical distributions negatively impacts the model’s ability to accommodate new classes.

Effect of the Joint Prototype Similarity Calculation. Furthermore, we evaluate the effectiveness of JPS in estimating class similarities by substituting it with the confusion matrix (CM) and buffer prototypes (BP) under the B0 5 steps, B0 10 steps, and B50 10 steps settings on the CIFAR100 dataset. The results are illustrated in Figure 5. Remarkably, both JPS and BP outperform the confusion matrix (CM) under B0 20 steps and B0 10 steps, with JPS further enhancing the performance of BP by integrating evolved historical prototype, resulting in improvements of 0.53% and 0.35% for the B0 20 steps and B0 10 steps settings, respectively. Notably, under the B50 10 steps setting, the performance of the BP is slightly lower than CM. This is primarily due to the limitation that only 20 samples per class can be stored in the buffer under this setting, which leads to insufficient prototype estimation from the buffer samples. In contrast, JPS effectively enhances the performance of CM. Overall, our proposed JPS achieves the highest accuracy across all experiments, with the mixing coefficient η set to 0.2.

4.4 Further Analysis

Analysis of Small Buffer Size. To further investigate the impact of buffer size on model performance, we conducted experiments with a fixed buffer size of 1000 exemplars. As shown in Table 4, FDDR outperforms MAFDRC across all settings and achieves performance

comparable to the model expansion approach DER. These results highlight the robustness of FDDR in memory-constrained scenarios.

Table 4. Performance on the CIFAR100 and ImageNet100 datasets under the B0 10 steps setting with a limited buffer size of $\mathcal{M} = 1k$.

Methods	CIFAR100			ImageNet100		
	5 steps	10 steps	20 steps	5 steps	10 steps	20 steps
DER w/o P	<u>74.37</u>	73.07	72.57	79.67	76.90	75.54
FOSTER B4	73.04	71.16	67.86	75.58	72.99	71.41
FOSTER	70.34	69.63	66.92	72.88	71.55	70.48
MAFDRC	73.46	71.62	67.91	<u>81.20</u>	<u>78.39</u>	72.37
FDDR	74.52	<u>72.63</u>	<u>68.97</u>	82.87	79.03	<u>73.75</u>

Visualization. We conduct visualization experiments to show that the KTDR strategy effectively generates diverse features for historical classes while preserving discriminative separability. By leveraging KTDR, we extract three feature types for each historical class: original, augmented, and sampled features. These features are visualized, with two classes shown per plot for clarity. As shown in Figure 4, KTDR enhances feature diversity and maintains clear decision boundaries, facilitating balanced model training.

5 Conclusion

In this paper, we first reveal the phenomenon of feature drift in class incremental learning, which leads to structure degradation of the historical distribution and consequently resulting in incomplete modeling of historical class distributions. Target this issue, we propose Feature Drift oriented Distribution Reconstruction framework, called FDDR, which utilizes both prior historical knowledge and new knowledge to reconstruct the complete historical distribution for balanced training. FDDR is based on three powerful strategies: GCR aligns historical distributions between the current and previous feature spaces to preserve structure information, JPS estimates inter-class similarity based on joint prototype cosine similarity, and KTDR enriches sparse historical distributions by transferring knowledge from similarity-weighted new class distributions. Extensive experiments verify the effectiveness of FDDR, demonstrating its superiority over state-of-the-art methods across various datasets.

Acknowledgements

This work is supported by the National Key RD Program of China (2022YFF0712100), NSFC (62276131), Natural Science Foundation of Jiangsu Province of China under Grant (BK20240081).

References

- [1] R. Aljundi, P. Chakravarty, and T. Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *CVPR*, pages 7120–7129, 2017.
- [2] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, volume 11207, pages 144–161, 2018.
- [3] R. Aljundi, M. Lin, B. Goujaud, and Y. Bengio. Gradient based sample selection for online continual learning. In *NeurIPS*, pages 11816–11825, 2019.
- [4] J. Bang, H. Kim, Y. Yoo, J. Ha, and J. Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *CVPR*, pages 8218–8227, 2021.
- [5] H. Chen, Y. Wang, and Q. Hu. Multi-granularity regularized rebalancing for class incremental learning. *TKDE*, 35(7):7263–7277, 2023.
- [6] J. Chen and B. Su. Instance-specific semantic augmentation for long-tailed image classification. *TIP*, 33:2544–2557, 2024.
- [7] X. Chen and X. Chang. Dynamic residual classifier for class incremental learning. In *ICCV*, pages 18697–18706, 2023.
- [8] X. Chen, Y. Zhou, D. Wu, W. Zhang, Y. Zhou, B. Li, and W. Wang. Imagine by reasoning: A reasoning-based implicit semantic data augmentation for long-tailed classification. In *AAAI*, pages 356–364, 2022.
- [9] Y. Choi, M. El-Khamy, and J. Lee. Dual-teacher class-incremental learning with data-free generative replay. In *CVPR*, 2021.
- [10] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.
- [11] J. Dong, W. Liang, Y. Cong, and G. Sun. Heterogeneous forgetting compensation for class-incremental learning. In *ICCV*, pages 11708–11717, 2023.
- [12] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, volume 12365, pages 86–102, 2020.
- [13] Z. Fu, K. Song, L. Zhou, and Y. Yang. Noise-aware image captioning with progressively exploring mismatched words. In *AAAI*, pages 12091–12099, 2024.
- [14] A. Gomez-Villa, D. Goswami, K. Wang, A. D. Bagdanov, B. Twardowski, and J. van de Weijer. Exemplar-free continual representation learning via learnable drift compensation. In *ECCV*, volume 15065, pages 473–490, 2024.
- [15] D. Goswami, A. Soutif-Cormerais, Y. Liu, S. Kamath, B. Twardowski, and J. van de Weijer. Resurrecting old classes with new data for exemplar-free continual learning. In *CVPR*, pages 28525–28534, 2024.
- [16] J. He. Gradient reweighting: Towards imbalanced class-incremental learning. In *CVPR*, pages 16668–16677, 2024.
- [17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
- [18] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin. Lifelong learning via progressive distillation and retrospection. In *ECCV*, volume 11207, pages 452–467, 2018.
- [19] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, and Y. Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *ICLR*, 2020.
- [20] T. Kim, J. Park, and B. Han. Cross-class feature augmentation for class incremental learning. In *AAAI*, pages 13168–13176, 2024.
- [21] J. Kirkpatrick, R. Pascanu, N. C. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796, 2016.
- [22] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.
- [23] M. Li, Z. Hu, Y. Lu, W. Lan, Y. Cheung, and H. Huang. Feature fusion from head to tail for long-tailed visual recognition. In *AAAI*, pages 13581–13589, 2024.
- [24] J. Liu, Y. Sun, C. Han, Z. Dou, and W. Li. Deep representation learning on long-tailed data: A learnable embedding augmentation perspective. In *CVPR*, pages 2967–2976, 2020.
- [25] Y. Liu, B. Schiele, and Q. Sun. Adaptive aggregation networks for class-incremental learning. In *CVPR*, pages 2544–2553, 2021.
- [26] Y. Ma, L. Jiao, F. Liu, S. Yang, X. Liu, and P. Chen. Geometric prior guided feature representation learning for long-tailed classification. *IJCV*, 132(7):2493–2510, 2024.
- [27] Z. Meng, X. Gu, Q. Shen, A. Tavares, S. Pinto, and H. Xu. H2T-FAST: head-to-tail feature augmentation by style transfer for long-tailed recognition. In *ECAI*, volume 372, pages 1712–1719, 2023.
- [28] A. K. Menon, S. Jayasumana, A. S. Rawat, H. Jain, A. Veit, and S. Kumar. Long-tail learning via logit adjustment. In *ICLR*, 2021.
- [29] C. Niu, G. Pang, and L. Chen. Graph continual learning with debiased lossless memory replay. In *ECAI*, volume 392, pages 1808–1815, 2024.
- [30] S. Park, Y. Hong, B. Heo, S. Yun, and J. Y. Choi. The majority can help the minority: Context-rich minority oversampling for long-tailed classification. In *CVPR*, pages 6877–6886, 2022.
- [31] S. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 5533–5542, 2017.
- [32] W. Shi, Y. Chen, Z. Zhao, W. Lu, K. Yan, and X. Du. Create and find flatness: Building flat training spaces in advance for continual learning. In *ECAI*, volume 372, pages 2138–2145, 2023.
- [33] F. Wang, D. Zhou, H. Ye, and D. Zhan. FOSTER: feature boosting and compression for class-incremental learning. In *ECCV*, volume 13685, pages 398–414, 2022.
- [34] S. Wang, Y. Zhan, Y. Luo, H. Hu, W. Yu, Y. Wen, and D. Tao. Joint input and output coordination for class-incremental learning. In *IJCAI*, pages 5108–5116, 2024.
- [35] H. Wen, L. Pan, Y. Dai, H. Qiu, L. Wang, Q. Wu, and H. Li. Class incremental learning with multi-teacher distillation. In *CVPR*, pages 28443–28452, 2024.
- [36] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu. Large scale incremental learning. In *CVPR*, pages 374–382, 2019.
- [37] S. Yan, J. Xie, and X. He. DER: dynamically expandable representation for class incremental learning. In *CVPR*, pages 3014–3023, 2021.
- [38] Y. Yang, D. Zhou, D. Zhan, H. Xiong, and Y. Jiang. Adaptive deep models for incremental learning: Considering capacity scalability and sustainability. In *SIGKDD*, pages 74–82, 2019.
- [39] Y. Yang, Y. Huang, W. Guo, B. Xu, and D. Xia. Towards global video scene segmentation with context-aware transformer. In *AAAI*, pages 3206–3213, 2023.
- [40] Y. Yang, Z. Sun, H. Zhu, Y. Fu, Y. Zhou, H. Xiong, and J. Yang. Learning adaptive embedding considering incremental class. *TKDE*, 35(3):2736–2749, 2023.
- [41] Y. Yang, Y. Zhang, X. Song, and Y. Xu. Not all out-of-distribution data are harmful to open-set active learning. In *NeurIPS*, 2023.
- [42] Y. Yang, D. Zhou, D. Zhan, H. Xiong, Y. Jiang, and J. Yang. Cost-effective incremental deep model: Matching model capacity with the least sampling. *TKDE*, 35(4):3575–3588, 2023.
- [43] Y. Yang, F. Wan, Q. Jiang, and Y. Xu. Facilitating multimodal classification via dynamically learning modality gap. In *NeurIPS*, 2024.
- [44] X. Yin, X. Yu, K. Sohn, X. Liu, and M. Chandraker. Feature transfer learning for face recognition with under-represented data. In *CVPR*, pages 5704–5713, 2019.
- [45] L. Yu, B. Twardowski, X. Liu, L. Herranz, K. Wang, Y. Cheng, S. Jui, and J. van de Weijer. Semantic drift compensation for class-incremental learning. In *CVPR*, pages 6980–6989, 2020.
- [46] B. Zhao, X. Xiao, G. Gan, B. Zhang, and S. Xia. Maintaining discrimination and fairness in class incremental learning. In *CVPR*, pages 13205–13214, 2020.
- [47] Q. Zhao, C. Jiang, W. Hu, F. Zhang, and J. Liu. MDCS: more diverse experts with consistency self-distillation for long-tailed recognition. In *ICCV*, pages 11563–11574, 2023.
- [48] Q. Zhao, Y. Dai, S. Lin, W. Hu, F. Zhang, and J. Liu. LTRL: boosting long-tail recognition via reflective learning. In *ECCV*, volume 15125, pages 1–18, 2024.
- [49] B. Zheng, D. Zhou, H. Ye, and D. Zhan. Multi-layer rehearsal feature augmentation for class-incremental learning. In *ICML*, 2024.
- [50] D. Zhou, Y. Yang, and D. Zhan. Learning to classify with incremental new class. *TNNLS*, 33(6):2429–2443, 2022.
- [51] D. Zhou, Q. Wang, H. Ye, and D. Zhan. A model or 603 exemplars: Towards memory-efficient class-incremental learning. In *ICLR*, 2023.
- [52] D. Zhou, Q. Wang, Z. Qi, H. Ye, D. Zhan, and Z. Liu. Class-incremental learning: A survey. *TPAMI*, 46(12):9851–9873, 2024.
- [53] Y. Zhou, J. Yao, F. Hong, Y. Zhang, and Y. Wang. Balanced destruction-reconstruction dynamics for memory-replay class incremental learning. *TIP*, 33:4966–4981, 2024.
- [54] J. Zhu, Z. Wang, J. Chen, Y. P. Chen, and Y. Jiang. Balanced contrastive learning for long-tailed visual recognition. In *CVPR*, pages 6898–6907, 2022.