IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 February 2024, accepted 14 March 2024, date of publication 18 March 2024, date of current version 22 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3378606

RESEARCH ARTICLE

MixER: Mixup-Based Experience Replay for Online Class-Incremental Learning

WON-SEON LIM[1], YU ZHOU[2], (Member, IEEE), DAE-WON KIM[1], (Member, IEEE), AND JAESUNG LEE[3]

[1]School of Computer Science and Engineering, Chung-Ang University, Dongjak-Gu, Seoul 06974, Republic of Korea
[2]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
[3]Department of Artificial Intelligence, Chung-Ang University, Dongjak-Gu, Seoul 06974, Republic of Korea

Corresponding authors: Dae-Won Kim (dwkim@cau.ac.kr) and Jaesung Lee (curseor@cau.ac.kr)

This work was supported in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) Grant funded by Korean Government [Ministry of Science and ICT (MSIT)] [Artificial Intelligence Graduate School Program (Chung-Ang University)] under Grant 2021-0-01341, in part by the National Research Foundation of Korea (NRF) Grant funded by Korean Government (MSIT) under Grant 2023R1A2C1006745, and in part by Shenzhen Fundamental Research Program under Grant JCYJ20220810112354002.

ABSTRACT Continual learning in the online class-incremental setting aims to learn new classes continuously from a consistent data stream while retaining the knowledge of old classes to prevent catastrophic forgetting. Traditional replay-based methods store and use old-class data to achieve this. However, they often overlook the representation shift caused by the incoming data streams, which leads to suboptimal classification accuracy. In this study, we propose a solution for mitigating representation shifts by incorporating asymmetric mixup training into the replay method. Our approach is based on the concept that mixup-based training enhances the stability of model predictions and gradient norms between training samples. Our method differs from typical mixup augmentation, which is uniformly applied to all data. Instead, it selectively targets the old data stored in the memory buffer, deliberately excluding the classes from the newly incoming data. This approach enables the model to learn new data while preserving the representation of the old data. Moreover, our experiments demonstrate the effectiveness of the proposed method, which not only enhances the performance of replay-based methods but can also be seamlessly integrated as an additional compatible module into various replay-based techniques. Evaluation on the CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets demonstrates that our approach surpasses existing replay-based methods. It addresses the limitations of conventional replay techniques and offers a potential solution for continual learning scenarios. Our source code is publicly available at https://github.com/laymond1/MixER.

INDEX TERMS Continual learning, online learning, replay-based learning, mixup training.

I. INTRODUCTION

Deep learning systems have achieved superior performance compared with humans in numerous computer vision applications [1], [2], [3]. However, unlike humans, these systems cannot accumulate knowledge over time; thus, they must commence training anew when learning new data. To solve this problem, continuous learning (CL) studies have been conducted to enable learning even when new data are continuously input [4], [5], [6]. The main challenge in CL is maintaining the knowledge learned in the past when learning new information. Early CL studies focused on task-incremental learning scenarios in which classes to be learned were divided by task, and learned offline [10], [11], [12]. However, this scenario differs significantly from the real world, where there are no distinct boundaries between tasks, and learning occurs continuously online [13]. Therefore, this study focuses on an online class-incremental learning scenario where there is no division of tasks and stream data are learned only once.

There are three main continual learning approaches: regularization [14], [15], [16], parameter isolation [17],

The associate editor coordinating the review of this manuscript and approving it for publication was Ganesh Naik.

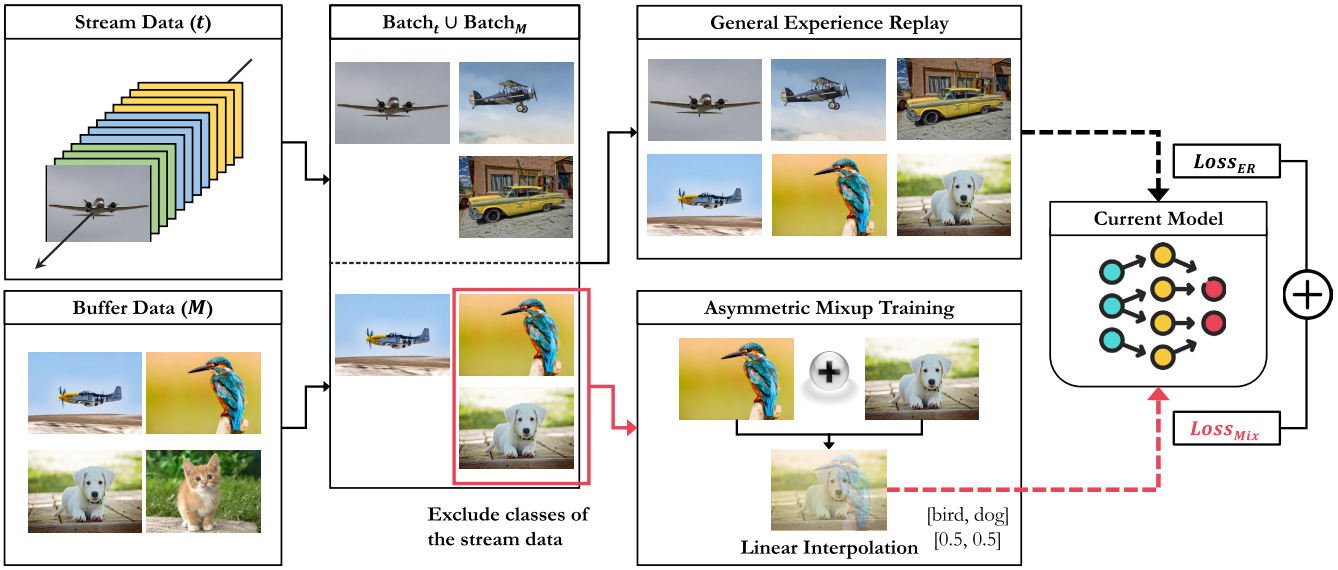


FIGURE 1. Overview of the Mixup-based experience replay (MixER). We receive a batch of stream data and sample a batch of buffer data according to a predefined sampling strategy. After composing the data, in general experience replay, the stream and the buffer batch data are used to train the model. However, our proposed asymmetric mixup training module only uses the buffer data and excludes classes from the stream data. This exclusion process removes labels corresponding to the classes in the stream batch data from the buffer batch data. These data points are linearly interpolated to create mixed-image pairs, thus facilitating the generalization of learned classes.

[18], [19], and replay-based learning methods [22], [23], [24], [25]. Among them, replay-based learning methods are simple and effective and have been widely used in online class-incremental learning studies. The replay method stores previously learned data samples in a memory buffer and utilizes them for training when new data samples are generated. However, this method is limited because it focuses solely on storing data samples and does not consider the previous representation shift in the model parameters caused by the large gradients of the new class data, as discussed in [26]. This limitation leads to a decrease in the classification performance.

In this study, we propose a method to mitigate the representation shift by applying asymmetric mixup augmentation to the replay method. This method is based on the concept that mixup-based training stabilizes model predictions and gradient norms between training samples [29], [30]. Specifically, we enhance the robustness of the model against representation shifts by asymmetrically applying mixup augmentation. Our method does not apply mixup augmentation to the incoming class data. Rather, it is applied to the old data in a memory buffer that does not belong to the classes of the incoming data. In this manner, the model learns new data while maintaining the old data representation. An overview of the proposed method is provided in FIGURE 1. Furthermore, experiments demonstrate that the proposed method improves the performance of replay-based methods and can be applied to various replay-based methods as an additional compatible module. The proposed method was evaluated on the CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets, and the results showed that it outperformed existing replay-based methods. The contributions of this study are as follows:

- To preserve the representation of the previously learned knowledge, we propose a method that applies an asymmetric mixup training module to the replay method.
- The proposed module can be applied to various replay-based methods as an additional compatible module.
- Extensive experiments on benchmark datasets of online class-incremental learning scenarios demonstrate that the proposed method outperforms existing replay-based methods.

II. RELATED WORK

Continual learning aims to design a model that can progressively acquire and build upon knowledge over time to adapt to continuously generated new data. In this context, the primary challenge in continual learning is preventing models from experiencing catastrophic forgetting (CF). Continual learning methods can be classified into three major categories based on the techniques employed.

The first category includes regularization-based methods that aim to mitigate CF by imposing constraints on the update of network parameters. For example, methods such as the EWC [14] introduce the fisher information-based regularization terms into the loss function to penalize the update of critical model parameters. Others, such as LwF [15], used their predictions of the old model to distill learned knowledge to mitigate prediction drift in old tasks. Recent work has shown that weight regularization techniques primarily used in continuous learning, such as EWC [14], SI [38], and MAS [11], are all related to the same theoretical quantity [16].

Next, parameter isolation methods aim to expand the neural network and mask certain parameters to prevent

TABLE 1. ER baseline methods for comparison.

Method	Training Strategy	Buffer Data
ER [22]	Utilize a memory buffer for retraining previous data.	$[x', y']$
DER [27]	Utilize a memory buffer for distilling previous logits.	$[x', z']$
DER++ [27]	Utilize a memory buffer for distilling previous logits and retraining previous data.	$[x', z', y']$

TABLE 2. Sample retrial methods of memory buffer.

Method	Sampling Strategy
Random [22]	Randomly extract samples from a memory buffer equal to the mini-batch size.
MIR [28]	Retrieve samples that have the most interference with the streaming data.
ASER [23]	Retrieve samples that are scored with Shapley value according to their ability to preserve previously learned classes.

forgetting. There are two main approaches within this category: the fixed architecture approach optimizes a binary mask to select specific parameters for each task [18], [20]. This approach ensures that previously learned tasks remain unchanged to avoid catastrophic forgetting. In this context, NISPA [19] proposed an architecture using a fixed-density sparse neural network with a stable path to preserve old knowledge and plastic paths that rewire connections to adapt to new tasks. The dynamic architecture approach, on the other hand, introduces novel parameters for new tasks while keeping the old parameters unchanged [21], [39].

The last category includes replay-based methods that utilize a memory buffer to store data from previous tasks, which can reduce CF. These methods retrieve samples from the memory buffer and combine them with the incoming data for model updates. Various strategies exist in this category, including experience replay (ER) [22], maximally interfered retrieval (MIR) [28], adversarial Shapley value experience replay (ASER) [23], gradient coreset replay (GCR) [24], and experience packing and replay (EPR) [25]. However, these methods primarily focus on data storage and may not fully consider the impact of new data, potentially leading to a decrease in classification performance.

Mixup augmentation has gained attention because of its potential to enhance model robustness and generalization. Mixup training, introduced by Zhang et al. [29], involves blending pairs of training samples and their corresponding labels to create new training examples. This technique encourages the model to make predictions that are linear interpolations of the original data, thereby providing smoother decision boundaries and reducing overfitting. Additionally, various augmentation methods are being employed, including the linear interpolation approach [33], [34], CutMix [31], SaliencyMix [32], and Co-Mixup [35]. These methods involve mixing patch images, utilizing a saliency map to mix informative patches, and optimizing the construction of mixup data to maximize individual data saliency and promote diversity among them. Mixup has been shown to be effective in improving the model performance and enhancing the generalization of CNNs [30], [40]. Moreover, Thulasidasan et al. [30] showed that Deep Neural Networks (DNNs) trained using a mixup exhibited reduced susceptibility to overly confident predictions when presented with out-of-distribution and random noise data, resulting in improved calibration. Thus, the integration of Mixup as both a regularizer and a strategic augmentation technique, as discussed in [36] and [37], demonstrates its effectiveness not only in improving accuracy and robustness to out-of-distribution data but also in simplifying the complexity involved in delineating optimal decision boundaries for improved model generalization.

In this study, we leverage the principles of mixup training to address the challenge of a representation shift in continual learning scenarios. Our approach applies mixup augmentation in an asymmetric manner to stabilize the predictions of the model and gradient norms while learning new data. This method offers a promising solution to mitigate the limitations of traditional replay-based learning methods, as discussed in [26].

III. PROPOSED METHOD

This study was motivated by the need to overcome the limitations of existing replay-based learning methods, which focus on data sample storage and overlook the representation shift in model parameters caused by the gradients of new class data. We propose an asymmetric mixup training approach to mitigate representation shifts. This enhances model stability and improves performance in online class-incremental learning scenarios. An overview of MixER is presented in FIGURE 1. In Section III-B we explain the general ER algorithm. Subsequently, in Section III-C, we discuss the details of our proposed method, namely, the asymmetric mixup module.

Algorithm 1 General Experience Replay

```
Input: Dataset D, Model Parameters θ, Learning rate η
Initialize Memory Buffer M, Parameters θ
while Data stream has not ended do
    Receive X_n ~ D_t
    Sample X_M ~ M   ▷ Retrieve sample according to the sampling strategy
    L = L_ER(X_n ∪ X_M)
    θ ← SGD(∇L, θ, η)   ▷ Parameters gradient update
    RESERVOIRUPDATE(M, X_n)
end while
```

A. PROBLEM DEFINITION

For continual learning, we focused on an online class-incremental continual learning scenario in which a model must continuously learn new classes from an online data stream. This data stream, denoted by $D = \{D_1, D_2, \ldots, D_N\}$, comprises samples X and their corresponding labels Y, where N represents the total number of tasks. There is no intersection among the classes across different tasks, implying that the sets D_i and D_j are separate when $i \neq j$. During the training phase, the data stream is processed sequentially, and the data of each task D_t are used to train the network for one epoch in task t. The objective of the model is to accurately predict all observed classes, which is the union of all classes observed up to the current task. Additionally, data distribution can change over time without explicit task notifications, making it challenging to adapt to evolving data patterns.

B. GENERAL EXPERIENCE REPLAY

Recent studies [41], [42], [43] have demonstrated that methods utilizing a memory buffer outperform regularization-based approaches in online continual learning scenarios. Therefore, we employed the replay method, which is commonly used as a standard baseline. The general ER method, as depicted in Algorithm 1, consists of a memory buffer and a model. After initializing the memory buffer and model parameters, replay training is conducted by receiving the streaming data. Replay learning involves extracting data from the memory buffer based on a sampling strategy and training it alongside the streaming batch data. Therefore, the loss function used for replay learning is $\mathcal{L}_{\text{ER}}(X_n \cup X_M)$. Parameter gradient updates are performed using this loss function, and the memory buffer is updated through reservoir sampling. This process is iterated at each step until the streaming data are exhausted.

C. ASYMMETRIC MIXUP MODULE

The motivation for the asymmetric mixup module is to stabilize the model's representation and to generalize the learned knowledge. Standard mixup training generates a mixed sample $\tilde{x}_{i,j}$ using a linear combination of samples x_i and x_j; thus, the model representation $f_\theta(\tilde{x}_{i,j})$ is defined as follows:

$$f_\theta(\tilde{x}_{i,j}) = f_\theta(\lambda \cdot x_i + (1-\lambda) \cdot x_j) \tag{1}$$

where $\lambda \sim Beta(\alpha, \alpha)$ and $\alpha \in (0, \infty)$ are the hyperparameters of the mixup training. In general, this can be summarized as follows: $f_\theta(\tilde{X})$

In our approach, we asymmetrically apply mixup training to the samples within the memory buffer during the training paradigm of the replay method.

$$\mathcal{L}_{\text{Mix}}^{\text{Asym}}(X_{\mathcal{M}}) = \mathcal{L}(\tilde{X}_{\mathcal{M}}, C_{\text{old}}) \tag{2}$$

$$= -\sum_{\tilde{x} \in \tilde{X}_{\mathcal{M}}^{\text{old}}} \log \frac{\exp\left(w_{c(\tilde{x})}^T f_\theta(\tilde{x})\right)}{\sum_{c \in C_{\text{all}}} \exp\left(w_c^T f_\theta(\tilde{x})\right)} \tag{3}$$

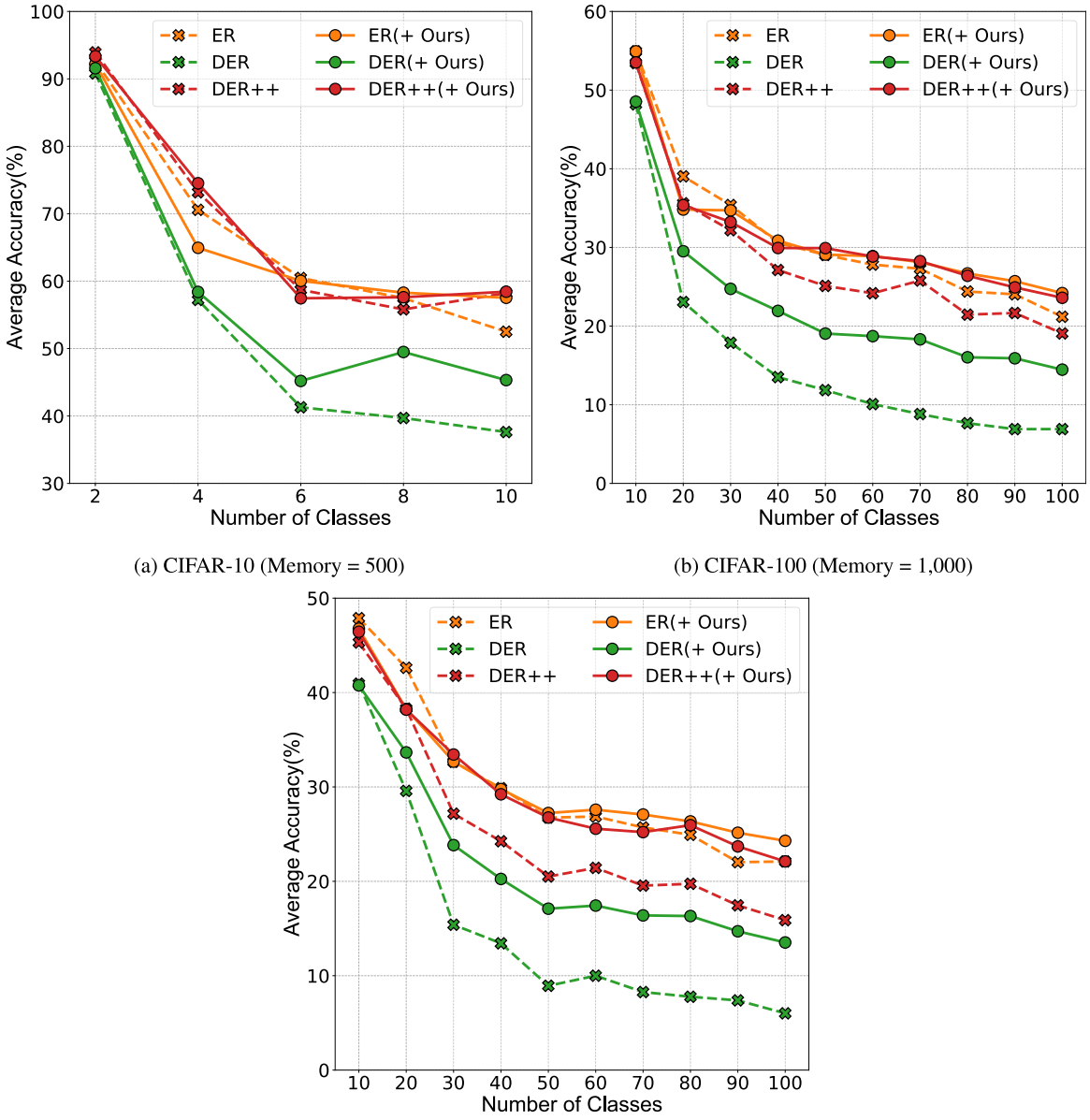
where C_{old} represents the set of old classes, $\tilde{X}_{\mathcal{M}}^{\text{old}}$ denotes the set of memory buffer data corresponding to the old classes, w_c represents the weight vector associated with the class c in the model's classification layer, and $c(x)$ and $c(\tilde{x})$ denote the label of x and the mixed class label of the mixed sample $\tilde{x}$, respectively. Since this equation applies operations such as softmax exclusively to the data of old classes, our asymmetric mixup module can generalize the model's prediction within the weight domain of the old classes. The entire process of using mixup training is applied to the general experience replay method, which is added as an additional component to the optimization term $\mathcal{L}_{\text{ER}}$. Thus, the loss function used for training is $\mathcal{L}_{\text{ER}}(X_n \cup X_{\mathcal{M}}) + \gamma \mathcal{L}_{\text{Mix}}^{\text{Asym}}(X_{\mathcal{M}})$, where γ is the hyperparameter of mixup training. The model parameters are updated using this loss function, and the remaining processes are performed in the same manner as in the general experience replay method.

IV. EXPERIMENTAL RESULTS

We conducted experiments to validate the performance of the proposed mixed augmentation-based module, and for this purpose, we compared the performance results based on the existing replay-based methods.

A. BENCHMARK DATASETS

We conducted experiments using the Sequential CIFAR-10 [44], Sequential CIFAR-100 [44], and Sequential Tiny-ImageNet [45] datasets, which are representative datasets reconfigured for online continual learning scenarios. The Sequential CIFAR-10 dataset is divided into 5 tasks, each involving the learning of 2 classes. The Sequential CIFAR-100 dataset is divided into 10 tasks, with each task involving the learning of 10 classes. Both datasets have an image resolution of 32 × 32 pixels, and out of a total of 60,000 samples, 50,000 are used for training, while the remaining 10,000 samples are used for testing. The Sequential Tiny-ImageNet dataset is divided into 10 tasks, with each task involving the learning of 10 classes. The dataset has an image resolution of 64 × 64 pixels. Our of a total of 55,000 samples, 50,000 are used for training, whereas the remaining 5,000 samples are used for testing.



(a) CIFAR-10 (Memory = 500)

(b) CIFAR-100 (Memory = 1,000)

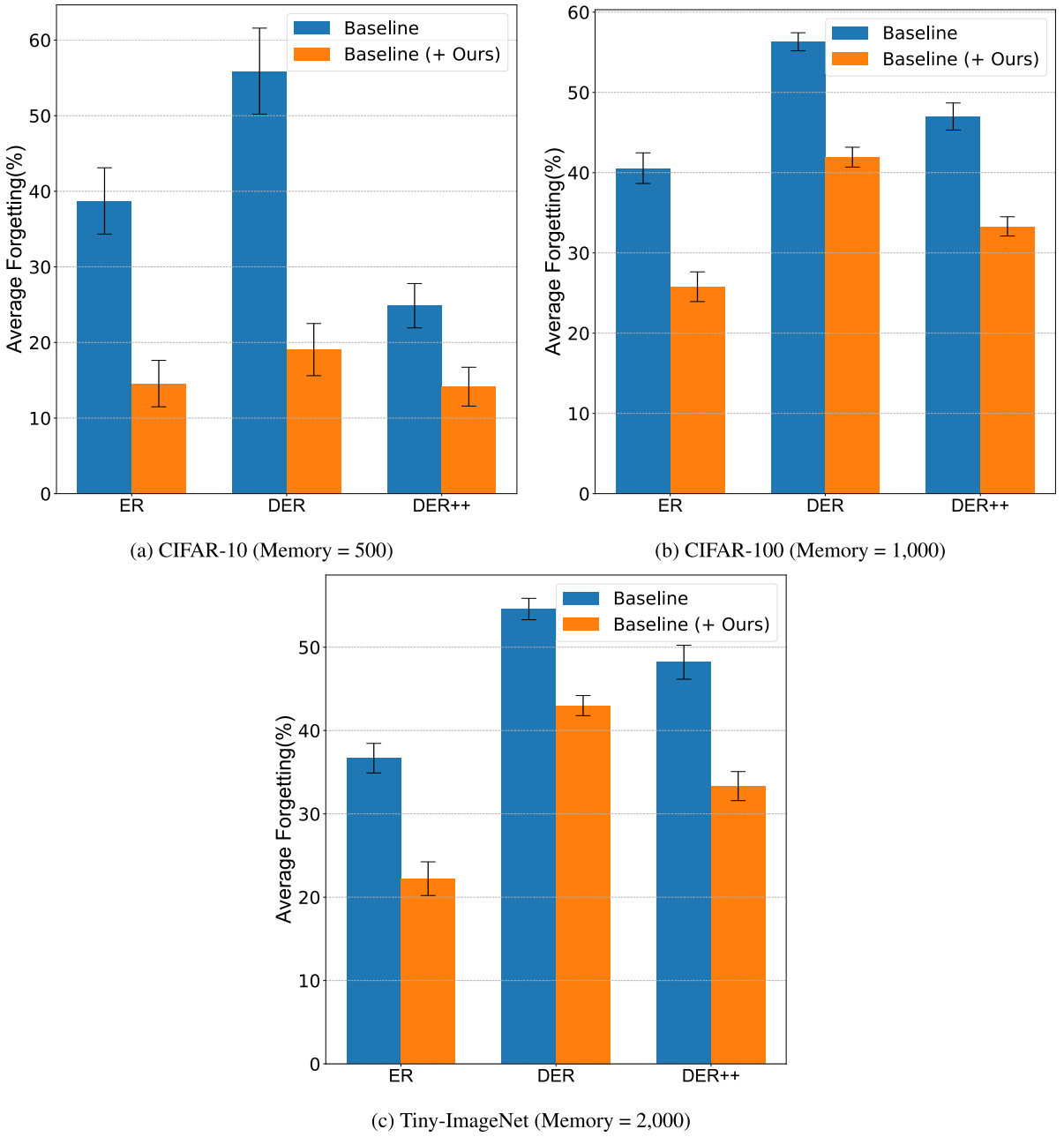
(c) Tiny-ImageNet (Memory = 2,000)

FIGURE 2. Comparison of average accuracy between the ER baselines and the proposed method across CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets.

B. EXPERIMENTAL SETTINGS

The experiments were conducted under the following conditions: The CPU used is an Intel(R) Core i9-10900X processor, and the GPU is an NVIDIA GeForce RTX 3090 with 24GB of memory. The backbone neural network used to train the algorithm is ResNet18 [1] with a learning rate of 0.01, and training is conducted using the stochastic gradient descent (SGD) optimizer. Both the learning rate and the SGD optimizer were selected through grid search from the search spaces of {0.1, 0.01, 0.001} for learning rates and {SGD, Adam [7], Adagrad [8], RMSProp [9]} for optimizers, respectively. Details of the optimizer settings are described in Appendix. The hyperparameters α and γ of our method are set using grid searches at {0.1, 0.5, 1, 5} and {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}, respectively. Specifically, α are set to 5 for CIFAR-10, 0.5 for CIFAR-100, and 0.5 for Tiny-ImageNet, whereas γ are set to 1. Each algorithm are configured to learn data with a batch size of 10 online, and the batch size of the data extracted from the memory buffer are set to 10, following the guidelines in [23] and [28].

In our evaluation, we focus on replay-based methods that operate effectively by using simple techniques in an online continual learning scenario. To ensure a fair comparison among these replay-based methodologies, we standardize memory buffer management using reservoir sampling [46] as the memory update strategy and random sampling [22] as the sample retrieval strategy. Following this approach, we establish the following representative experience replay (ER)



(a) CIFAR-10 (Memory = 500)

(b) CIFAR-100 (Memory = 1,000)

(c) Tiny-ImageNet (Memory = 2,000)

FIGURE 3. Comparison of average forgetting between the ER baselines and the proposed method across different datasets, including CIFAR-10, CIFAR-100, and Tiny-ImageNet.

methodologies, as specified in Table 1, to serve as the baseline methods for applying the proposed modules. In addition, we establish the representative sample retrieval methodologies specified in Table 2 to validate our approach across the various retrieval methods.

To assess the performance of each algorithm, we employ two key performance metrics that are widely used in the field of continual learning: average accuracy and average forgetting [47]. The average accuracy measures the overall performance across completed tasks in a continual learning scenario, whereas the average forgetting quantifies the extent to which the algorithm has forgotten information about previously completed tasks. The average accuracy can be defined as follows:

$$\text{Average Accuracy}(\text{A}_\text{T}) = \frac{1}{T}\sum_{j=1}^{T} a_{T,j},$$

And the average forgetting can be defined as follows:

$$\text{Average Forgetting}(\text{F}_\text{T}) = \frac{1}{T-1}\sum_{j=1}^{T-1} f_{T,j} \tag{4}$$

$$\text{where } f_{i,j} = \max_{l\in 1,\ldots,i-1} a_{l,j} - a_{i,j}$$

TABLE 3. Experimental results of Average Accuracy (↑ is better) on CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets, with M denoting the memory buffer size. All results are reported in the form of mean ± standard deviation for ten runs. ↑ indicates that the proposed module has a statistically significant improvement over the baseline method, as determined by a paired t-test with a p-value < 0.05.

Dataset	M	ER	**ER (+ Ours)**	DER	**DER (+ Ours)**	DER++	**DER++ (+ Ours)**
CIFAR-10	$0.2K$	43.29 ± 2.61	**48.87 ± 2.07**(↑)	36.96 ± 3.82	**44.25 ± 1.85**(↑)	50.11 ± 2.27	**53.22 ± 2.82**(↑)
	$0.5K$	52.50 ± 3.07	**57.53 ± 1.26**(↑)	37.61 ± 3.49	**45.32 ± 3.55**(↑)	58.21 ± 1.07	58.44 ± 2.00
	$1.0K$	58.95 ± 3.24	60.59 ± 2.60	36.85 ± 2.83	**47.57 ± 2.38**(↑)	59.74 ± 3.39	60.31 ± 4.43
CIFAR-100	$0.5K$	17.03 ± 1.34	**19.32 ± 1.06**(↑)	6.82 ± 0.22	**13.45 ± 1.10**(↑)	16.20 ± 1.38	**18.99 ± 1.53**(↑)
	$1.0K$	21.18 ± 1.43	**24.19 ± 1.07**(↑)	6.90 ± 0.17	**14.46 ± 1.19**(↑)	19.07 ± 1.31	**23.60 ± 0.94**(↑)
	$2.0K$	24.35 ± 1.03	**26.29 ± 1.09**(↑)	6.79 ± 0.25	**14.37 ± 0.91**(↑)	19.43 ± 1.09	**26.34 ± 1.30**(↑)
Tiny-ImageNet	$1.0K$	19.75 ± 1.20	**21.04 ± 1.60**(↑)	5.93 ± 0.30	**14.63 ± 0.63**(↑)	15.65 ± 1.26	**20.12 ± 0.97**(↑)
	$2.0K$	22.10 ± 1.33	**24.29 ± 1.13**(↑)	6.01 ± 0.24	**13.52 ± 0.40**(↑)	15.88 ± 0.71	**22.11 ± 1.22**(↑)
	$5.0K$	22.94 ± 0.78	**24.90 ± 2.16**(↑)	5.85 ± 0.32	**13.53 ± 0.89**(↑)	15.23 ± 1.12	**23.06 ± 1.51**(↑)

TABLE 4. Experimental results of Average Forgetting (↓ is better) on CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets, with M denoting the memory buffer size. All results are reported in the form of mean ± standard deviation for ten runs. ↓ indicates that the proposed module has a statistically significant decline over the baseline method, as determined by a paired t-test with a p-value < 0.05.

Dataset	M	ER	**ER (+ Ours)**	DER	**DER (+ Ours)**	DER++	**DER++ (+ Ours)**
CIFAR-10	$0.2K$	52.70 ± 2.82	**34.96 ± 2.77** (↓)	55.27 ± 5.92	**22.00 ± 6.32** (↓)	37.21 ± 4.46	**22.87 ± 2.47** (↓)
	$0.5K$	38.72 ± 4.39	**14.56 ± 3.07** (↓)	55.89 ± 5.69	**19.06 ± 3.45** (↓)	24.86 ± 2.93	**14.15 ± 2.57** (↓)
	$1.0K$	30.83 ± 4.54	**12.54 ± 2.13** (↓)	58.95 ± 4.52	**15.60 ± 3.92** (↓)	24.40 ± 6.77	**13.11 ± 3.39** (↓)
CIFAR-100	$0.5K$	45.22 ± 1.44	**33.38 ± 1.64** (↓)	56.48 ± 1.14	**41.92 ± 1.10** (↓)	49.44 ± 1.00	**40.83 ± 1.32** (↓)
	$1.0K$	40.56 ± 1.69	**25.79 ± 1.20** (↓)	56.31 ± 1.91	**41.93 ± 1.85** (↓)	46.99 ± 1.12	**33.30 ± 1.23** (↓)
	$2.0K$	37.82 ± 1.73	**23.63 ± 1.26** (↓)	56.48 ± 1.70	**42.99 ± 1.30** (↓)	47.20 ± 0.85	**31.95 ± 1.36** (↓)
Tiny-ImageNet	$1.0K$	38.21 ± 1.70	**25.06 ± 1.34** (↓)	54.30 ± 1.73	**40.64 ± 2.36** (↓)	48.38 ± 1.02	**33.96 ± 1.30** (↓)
	$2.0K$	36.68 ± 2.03	**22.22 ± 1.74** (↓)	54.58 ± 1.77	**42.99 ± 2.02** (↓)	48.20 ± 1.28	**33.33 ± 1.21** (↓)
	$5.0K$	36.46 ± 1.31	**22.85 ± 1.61** (↓)	54.48 ± 1.28	**43.44 ± 1.88** (↓)	49.21 ± 0.96	**33.09 ± 1.20** (↓)

where i represents the number of tasks used for training, j represents the number of tasks used for testing, $a_{i,j}$ denotes the accuracy achieved on task j after training up to task i, and $f_{i,j}$ represents the decrease in the accuracy of task j after training up to task i.

C. COMPARISON RESULTS

FIGURE 2 and 3 display the results comparing the average accuracy and forgetting between ER baseline and proposed methods in an online class-incremental scenario with sequentially increasing classes for the CIFAR-10, CIFAR-100, and Tiny-ImageNet datasets. In FIGURE 2, our method demonstrates more remarkable performance improvement in CIFAR-100 and Tiny-ImageNet compared to CIFAR-10. Furthermore, the baseline methods exhibit performance decreases as the new classes are gradually learned, whereas the results of applying our method demonstrate a significantly smoother decline in performance. This indicates that our method functions effectively in an online class-incremental learning scenario and is more effective in long-sequential tasks than short-sequential tasks. This is because early learned representations are more easily forgotten in long-sequential tasks, but our method better leverages previous representations. In particular, in FIGURE 2b (CIFAR-100) and 2c (Tiny-ImageNet), DER++ performed lower than ER, whereas after our method was applied, it performed higher than ER in CIFAR-100 and similar in Tiny-ImageNet.

FIGURE 3 shows the average forgetting performance when all stream data learning is completed. For all the benchmark datasets, our method effectively reduced the

number of forgetting baseline methods. In FIGURE 3a (CIFAR-10), after the application of our method, the forgetting performances of the ER and DER decreased by approximately 60% compared with the baselines, and the forgetting performance of DER++ decreased by approximately 43%. In addition, as shown in FIGURE 3b (CIFAR-100) and 3c (Tiny-ImageNet), our method reduced forgetting by approximately 36%, 22%, and 30% for ER, DER, and DER++, respectively. This shows that our method reduces the shift in previously learned knowledge compared with the baseline ER methods.

D. EFFECTS OF MEMORY BUFFER SIZE AND SAMPLING STRATEGY

To validate the performance of the proposed module from the perspective of the memory buffer size, we conducted additional experiments. In this experiment, we vary the memory buffer size, which is crucial in replay methods, to assess the performance of the proposed module under different memory buffer size conditions. TABLE 3 and 4 present the comparative results between baseline replay methods and the proposed module when applied with various memory buffer sizes. We conducted ten experimental runs and averaged the results. Furthermore, to analyze the results of applying the proposed module to baseline replay methods clearly, we conducted pairwise t-tests at a significance level of 95%.

As shown in Table 3, the proposed method demonstrates an improvement in the average accuracy across all memory buffer sizes. For CIFAR-10, the performance improvement of the proposed method was more noticeable when the memory buffer size was smaller. When the memory buffer size was 0.2 K, our method showed performance improvements of 13%, 20%, and 6% for ER, DER, and DER++, respectively. When it is 1 K, it exhibits performance improvements of 3% and 1% for ER and DER++, respectively, and 30% for DER. In CIFAR-100 and Tiny-ImageNet, increasing the memory buffer size for DER and DER++ did not result in significant performance improvements compared with ER. Specifically, when increasing the memory buffer size from 0.5 K to 2K and from 1K to 5K for DER++ in CIFAR-100 and Tiny-ImageNet, the performance improvements were 20 % and -3 %, respectively. However, with our method, DER++ showed a significant performance improvement of 39% on CIFAR-100 and a notable performance improvement of 15 % on Tiny-ImageNet.

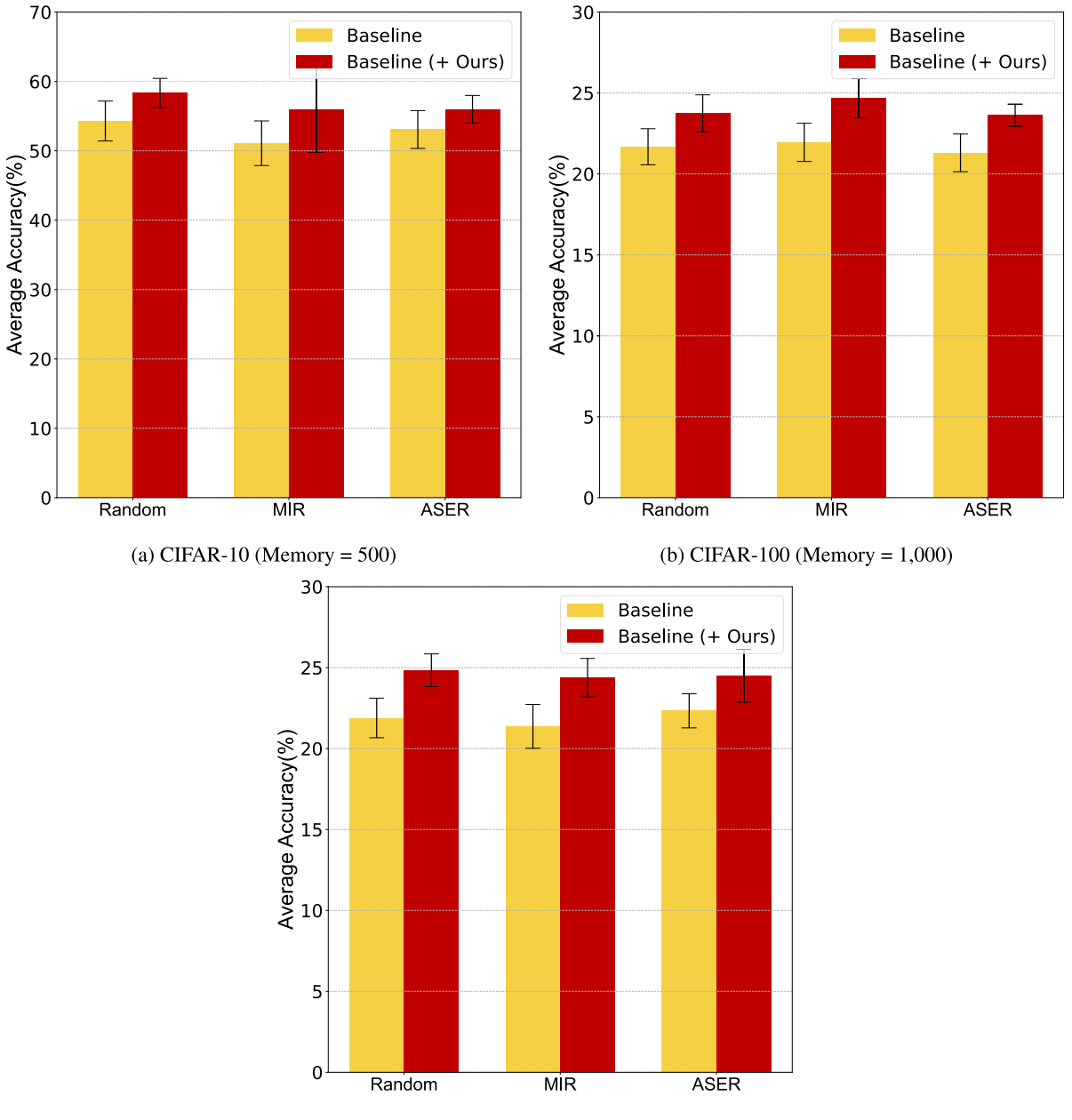
We also analyzed the average forgetting performance of each algorithm for varying memory buffer sizes. As indicated in Table 4, the proposed method effectively reduces forgetting across all memory buffer sizes and benchmark datasets. For example, for CIFAR-10, forgetting decreases, on average, by over 43%, whereas in CIFAR-100, the reduction is 25%, and in Tiny-ImageNet, it exceeds 22%. In CIFAR-10, unlike in the accuracy analysis, forgetting tended to decrease as the memory buffer size increased. When the memory buffer size increased from 0.2 K to 1 K, the forgetting performance of ER, DER, and DER++ applied with our method decreased by 64%, 29%, and 42%, respectively. In contrast, in CIFAR-100 and Tiny-ImageNet, methods other than ER did not show significant changes in the forgetting performance based on the memory buffer size.

To analyze the effect of the sampling strategy on the performance of the proposed module, we conducted experiments using different sampling strategies. Specifically, we compared the performance of our method when applied to random, maximal interference retrieval (MIR), and adversarial Shapley value experience replay (ASER) retrieval sampling. Figure 4 and 5 show the average accuracy and forgetting performance of each algorithm under different sampling strategies. As shown in Figure 4, our method leads to performance improvements over the baseline methods across all sampling strategies. Overall, the random sampling method performed the best in Figure 4a (CIFAR-10) and 4c (Tiny-ImageNet), with improvements of 7% and 14%, respectively. MIR sampling exhibited the most substantial performance improvement when our method was applied, despite having the lowest baseline performance among all the benchmark datasets. In Figure 4b (CIFAR-100), MIR sampling shows an increase of 13%, whereas in Figure 4c (Tiny-ImageNet) it shows a 14% improvement. In contrast, ASER demonstrated the smallest performance gains, increasing by 5% on CIFAR-10 and by approximately 10% on both CIFAR-100 and Tiny-ImageNet. Similarly, Figure 5 shows that forgetting decreases compared with the baseline methods when our method is applied across all sampling strategies. MIR sampling, as observed in Figure 5b (CIFAR-10) and 5c (Tiny-ImageNet), reduces forgetting to a level similar to that of the other sampling strategies, even when the baseline method exhibited the highest forgetting across all datasets. This demonstrates the synergistic effectiveness of the proposed method when combined with the MIR sampling strategy. Furthermore, we demonstrate the overall effectiveness of our method across various sampling strategies, illustrating its compatibility with and adaptability to the ER method.

E. EFFECT OF THE CLASS IMBALANCE

To verify the effectiveness of our method in the context of class imbalance, reflecting the more complex scenarios encountered in the real world, we reconfigured CIFAR-100 and Tiny-ImageNet into their imbalanced versions. We adopted a long-tailed imbalance setting [48], characterized by an exponential decay in sample sizes across different classes. This approach clearly distinguishes minority and majority classes, which is particularly useful for mirroring complex real-world situations.

Figure 6 demonstrates that the overall performance is lower compared to the class-balanced setting in Figure 2 due to the increased difficulty of the problem. We observed that our method encountered difficulties in learning the classes in the second and third tasks, as shown in both Figures 6a and 6b.



(a) CIFAR-10 (Memory = 500)

(b) CIFAR-100 (Memory = 1,000)

(c) Tiny-ImageNet (Memory = 2,000)

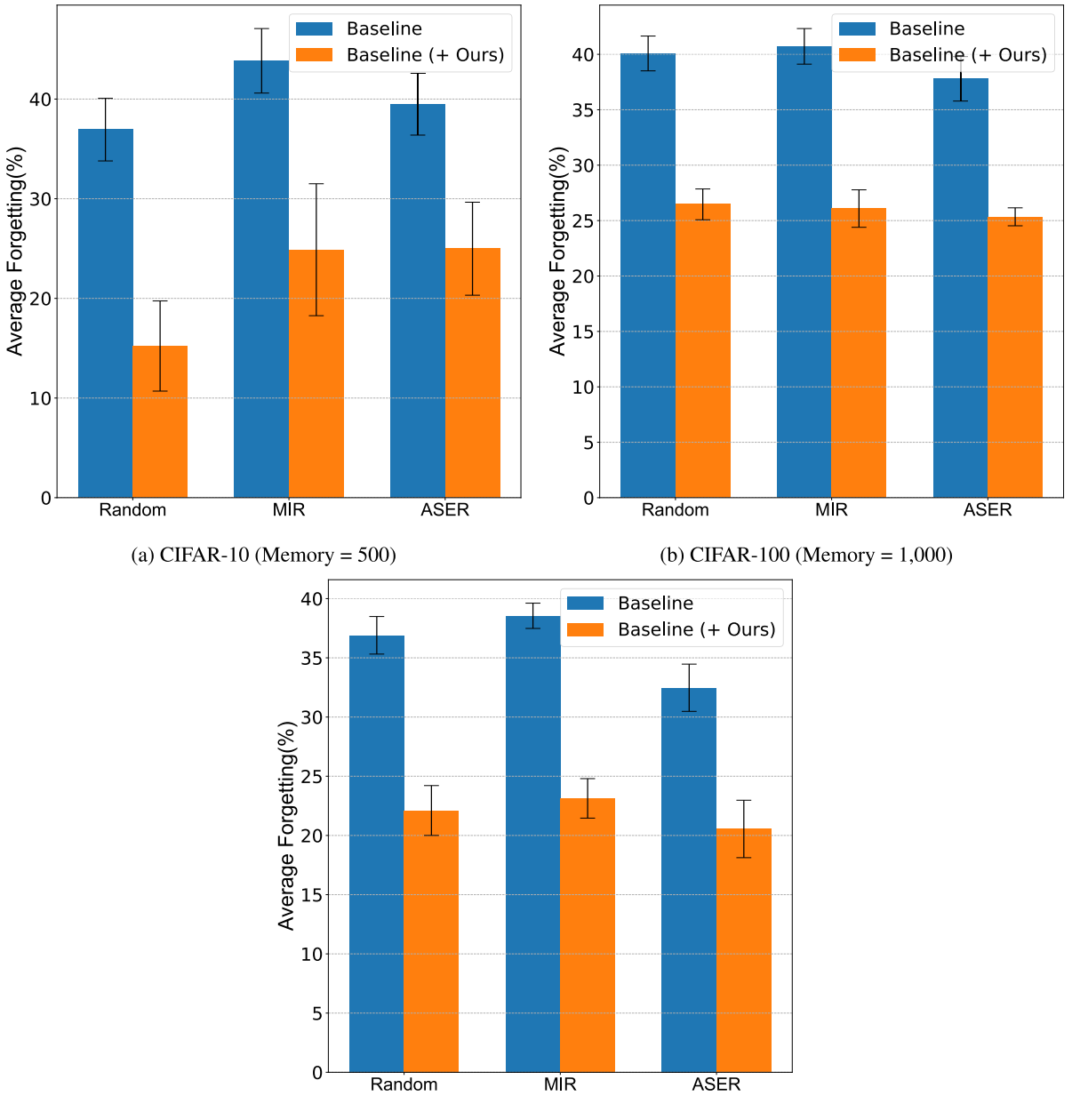
FIGURE 4. Comparison of average accuracy with different retrieval sampling methods across benchmark datasets, including CIFAR-10, CIFAR-100, and Tiny-ImageNet.

This suggests that the initial extreme class distribution in the early tasks could cause performance degradation when our asymmetric training module focuses on a minority of past classes. Nevertheless, as classes are incrementally added and learned, our method gradually mitigates this issue, eventually outperforming the baseline method from the fourth and fifth tasks. Furthermore, the performance degradation observed due to the class distribution in these initial stages represents a limitation of our method and merits further investigation as potential future work.

We also compare the average forgetting performance on the imbalanced CIFAR-100 and Tiny-ImageNet datasets, as shown in Figure 7. After applying our method, we observed an average reduction in forgetting of 30% for CIFAR-100 and 33% for Tiny-ImageNet. These results are comparable to those reported in Table 4 when employing a buffer of identical size with balanced datasets. Therefore, our method demonstrates its ability to preserve learned knowledge in both class balance and class imbalance settings with respect to forgetting.

F. ABLATION STUDY

We conducted a comparative analysis of various mixup training and augmentation strategies to demonstrate the



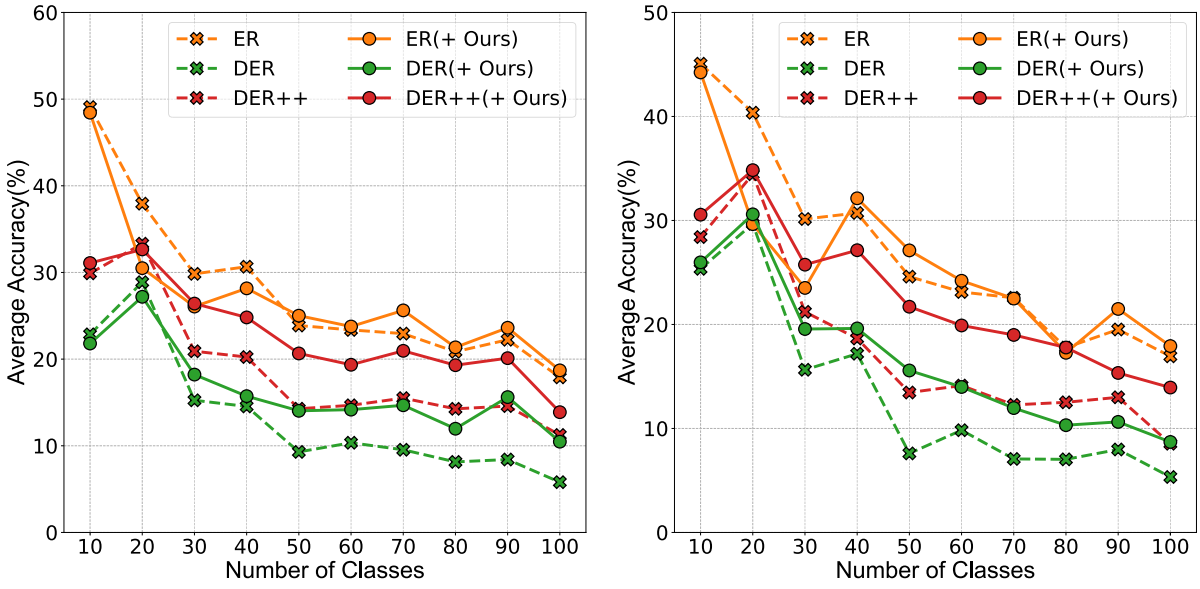
(a) CIFAR-10 (Memory = 500)

(b) CIFAR-100 (Memory = 1,000)

(c) Tiny-ImageNet (Memory = 2,000)

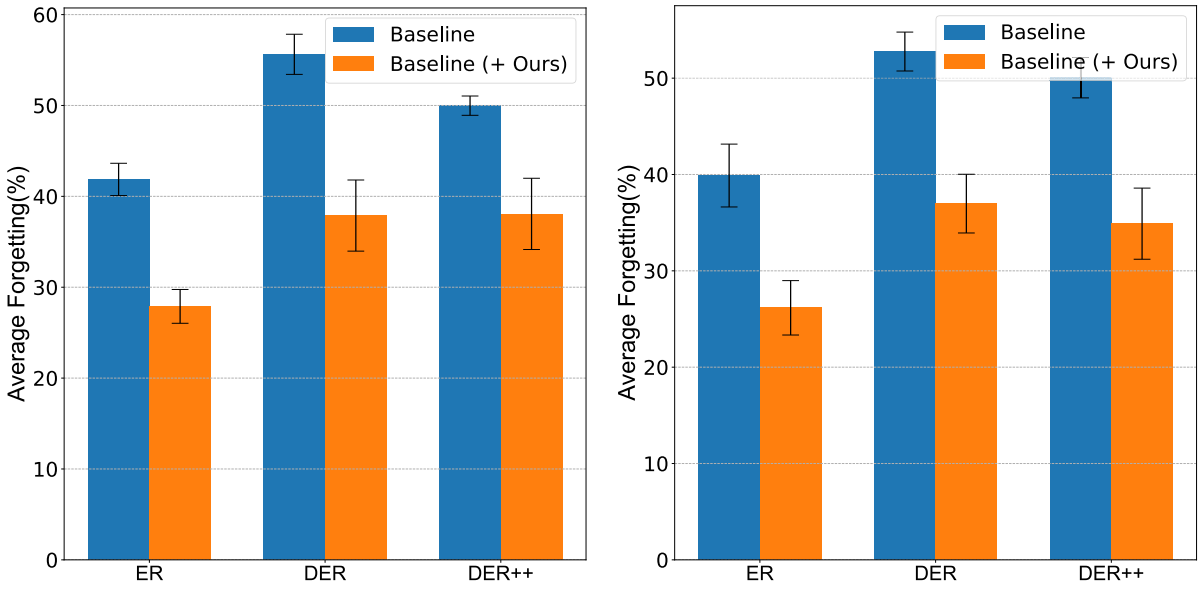
FIGURE 5. Comparison of average forgetting with different retrieval sampling methods across benchmark datasets, including CIFAR-10, CIFAR-100, and Tiny-ImageNet.

effectiveness of the proposed asymmetric mixup training method and the mixup augmentation technique employed. As illustrated in Table 5, our method, when applying asymmetric mixup training, exhibited superior performance compared to the other mixup training methods. The analysis of the forgetting performance reveals a significant reduction when we employ asymmetric mixup training, surpassing other methods. This suggests that our method not only significantly improves accuracy performance compared to other methods, but also substantially reduces the occurrence of forgetting. This is attributed to our method's emphasis on preserving previously learned representations by exclusively utilizing data from old classes ($X_{\mathcal{M}}, C_{\text{old}}$), as described in Eq. (2). In contrast, mixup training with streaming batch data (X_n), which focuses on newly incoming stream data, underperforms compared to the baseline ER method. Furthermore, methods that include a portion of the classes from the streaming data exhibit lower performance relative to our method. Additionally, unlike other methods, our approach selectively updates only the old class data, resulting in more efficient utilization of computational resources during training. Consequently, when there are no old class data in the sampled data, the same computational resources as in the baseline ER method are employed, with



(a) Imbalanced CIFAR-100 (Memory = 1,000) (b) Imbalanced Tiny-ImageNet (Memory = 2,000)

FIGURE 6. Comparison of average accuracy between the ER baselines and the proposed method across imbalanced CIFAR-100, and Tiny-ImageNet datasets.



(a) Imbalanced CIFAR-100 (Memory = 1,000) (b) Imbalanced Tiny-ImageNet (Memory = 2,000)

FIGURE 7. Comparison of average forgetting between the ER baselines and the proposed method across different datasets, including imbalanced CIFAR-100, and Tiny-ImageNet.

maximum utilization occurring when all sampled data fall under the old class. As indicated in Table 5, our method is appropriate for online learning with lower levels of GFLOPs.

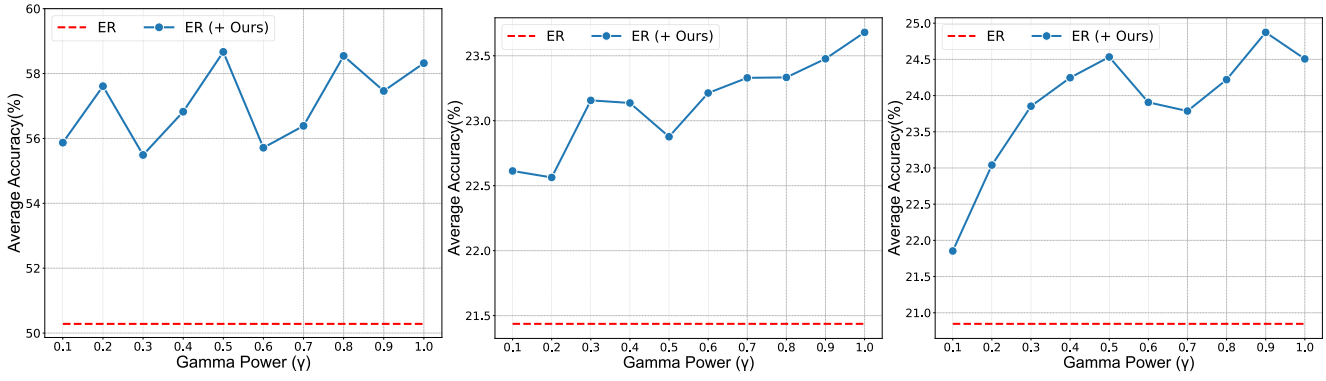
Table 6 displays the performance results of each augmentation implemented in the proposed module on the CIFAR-10 dataset. The results indicate that LinearMix, which directly blends input images, is more effective than ManifoldMix, which integrates features within the manifold domain despite employing linear interpolation methods. Additionally, the LinearMix method demonstrated superior average accuracy and forgetting performance compared to patch image mixup methods such as CutMix and SaliencyMix. In particular, the LinearMix method exhibited an average of 3% to 5% lower forgetting than other methods. We believe this is because the LinearMix method, by employing a linear combination of input images, results in less information loss of the original image compared to techniques that involve cutting the input image to combine patches. Therefore, the LinearMix method, in combination with our asymmetric

TABLE 5. Experimental results of Average Accuracy and Forgetting on CIFAR-10 with different mixup training methods.

Method	Train GFLOPs	M=0.2K		M=0.5K		M=1k	
		Avg. Acc.(↑)	Avg. Forget. (↓)	Avg. Acc.(↑)	Avg. Forget. (↓)	Avg. Acc.(↑)	Avg. Forget. (↓)
$\mathcal{L}_{\text{ER}}(X_n \cup X_{\mathcal{M}})$	44.63	43.29 ± 2.61	52.70 ± 2.82	52.50 ± 3.07	38.72 ± 4.39	58.95 ± 3.24	30.83 ± 4.54
$+ \mathcal{L}_{\text{Mix}}(X_n \cup X_{\mathcal{M}})$	89.26	47.14 ± 3.64	47.36 ± 4.82	57.53 ± 2.37	33.24 ± 3.86	59.94 ± 2.13	30.54 ± 3.32
$+ \mathcal{L}_{\text{Mix}}(X_n)$	66.95	34.63 ± 2.21	62.22 ± 2.83	41.97 ± 3.96	54.27 ± 5.17	45.91 ± 4.37	50.20 ± 5.32
$+ \mathcal{L}_{\text{Mix}}(X_{\mathcal{M}})$	66.95	48.08 ± 2.29	41.14 ± 4.67	56.69 ± 3.51	28.51 ± 5.48	59.17 ± 4.04	27.64 ± 5.15
$+ \mathcal{L}_{\mathbf{Our}}(\mathbf{X}_{\mathcal{M}}, \mathbf{C}_{\text{old}})$	**(44.63, 66.95)**	$\mathbf{48.99 \pm 3.26}$	$\mathbf{34.26 \pm 4.02}$	$\mathbf{57.70 \pm 2.39}$	$\mathbf{15.68 \pm 3.56}$	$\mathbf{61.34 \pm 1.47}$	$\mathbf{11.22 \pm 2.44}$

TABLE 6. Experimental results of Average Accuracy and Forgetting on CIFAR-10 with different mixup augmentation methods.

Method	M=0.2K		M=0.5K		M=1k	
	Avg. Acc.(↑)	Avg. Forget. (↓)	Avg. Acc.(↑)	Avg. Forget. (↓)	Avg. Acc.(↑)	Avg. Forget. (↓)
CutMix [31]	45.11 ± 2.62	38.15 ± 5.59	57.27 ± 1.12	18.20 ± 4.21	58.21 ± 2.76	16.92 ± 4.43
SaliencyMix [32]	43.80 ± 7.06	39.26 ± 6.10	53.82 ± 5.20	19.60 ± 5.20	59.15 ± 2.98	16.27 ± 4.34
ManifoldMix [33]	42.04 ± 3.39	44.37 ± 6.14	56.04 ± 2.70	20.58 ± 4.13	59.25 ± 2.58	16.02 ± 3.76
LinearMix (Our)	$\mathbf{48.99 \pm 3.26}$	$\mathbf{34.26 \pm 4.02}$	$\mathbf{57.70 \pm 2.39}$	$\mathbf{15.68 \pm 3.56}$	$\mathbf{61.34 \pm 1.47}$	$\mathbf{11.22 \pm 2.44}$



(a) CIFAR-10 (Memory = 500) (b) CIFAR-100 (Memory = 1,000) (c) Tiny-ImageNet (Memory = 2,000)

FIGURE 8. Comparison of the performance between the baseline Experience Replay (ER) and the proposed method on the CIFAR-10 dataset under the online class incremental scenario. The coefficient gamma improves the performance, which indicates that our method is successfully adaptable to experience replay methods.

training approach, enhances the ability to preserve previously learned knowledge.

FIGURE 6 illustrates how performance changes when the proposed asymmetric mixup loss term is applied to the baseline ER method with varying values of the coefficient gamma. As shown in FIGURE 6, our method leads to performance improvements as the gamma coefficient increases when applied to ER across benchmark datasets. Specifically, with a gamma value of 0.1, the performance improved by 12%, 6%, and 5% for CIFAR-10, CIFAR-100, and Tiny-ImageNet, respectively, compared with ER. When gamma was set to 1.0, the performance improvements were 16%, 11%, and 18% for CIFAR-10, CIFAR-100, and Tiny-ImageNet, respectively, compared with ER. As gamma increases from 0.1 to 1.0, the performance improvements were 4%, 5%, and 12% for CIFAR-10, CIFAR-100, and Tiny-ImageNet, respectively. This demonstrates that the proposed asymmetric mixup loss term enhances the performance as gamma increases when applied to the ER.

V. CONCLUSION

In this study, we introduce a novel and effective module for online class-incremental continual learning that considers the stability of the model's representation and the generalization

TABLE 7. Average Accuracy (↑ is better) on CIFAR-10 with different optimizers and learning rates. All results are reported in the form of mean ± standard deviation for ten runs.

Optimizer	lr=0.1	lr=0.01	lr=0.001
SGD	**36.54 ± 3.80** (↑)	**57.30 ± 1.61** (↑)	46.20 ± 1.19
Adam	16.47 ± 6.42	28.33 ± 3.19	**56.78 ± 7.10** (↑)
Adagrad	16.59 ± 4.44	30.57 ± 2.98	31.45 ± 2.87
RMSProp	17.89 ± 2.94	26.50 ± 2.60	39.56 ± 8.24

of the learned data in the online data stream. Specifically, we adopted an asymmetric mixup training approach for the streaming and memory data. This asymmetric training approach strengthens the model's resilience to shifts in the representation of new classes and enhances the generalization of the acquired knowledge. Furthermore, the proposed module can be readily adapted to existing replay-based methods. Extensive experiments conducted on widely used benchmark datasets in online continual learning validated the effectiveness of our approach.

APPENDIX

OPTIMIZER SETTINGS

For clarity regarding the optimizer and learning rate choice in our experiments, we present the results in Table 7. After evaluating four optimizers at progressively reduced learning rates of 0.1, 0.01, and 0.001, we observed that the SGD optimizer delivered superior performance at both 0.1 and 0.01 learning rates. In contrast, the Adam optimizer was more effective at a learning rate 0.001. Considering the comprehensive results of the experiments, the SGD optimizer was selected for its consistent, robust performance across various tests. Specifically, an SGD with a learning rate of 0.01, which showcased the highest performance, was chosen.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 91–99.

[3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[4] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter, "Continual lifelong learning with neural networks: A review," *Neural Netw.*, vol. 113, pp. 54–71, May 2019.

[5] M. D. Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3366–3385, Jul. 2022.

[6] 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.

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[8] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. 61, pp. 2121–2159, 2011.

[9] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited*, vol. 14, no. 8, p. 2, 2012.

[10] S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, and B.-T. Zhang, "Overcoming catastrophic forgetting by incremental moment matching," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Dec. 2017, pp. 4655–4665.

[11] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 139–154.

[12] J. Serra, D. Suris, M. Miron, and A. Karatzoglou, "Overcoming catastrophic forgetting with hard attention to the task," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4548–4557.

[13] J. He, R. Mao, Z. Shao, and F. Zhu, "Incremental learning in online scenario," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2020, pp. 13926–13935.

[14] K. James, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.

[15] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. Torr, "Riemannian walk for incremental learning: Understanding forgetting and intransigence," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 532–547.

[16] F. Benzing, "Unifying importance based regularisation methods for continual learning," in *Proc. Int. Conf. Artif. Intell. Stat.*, vol. 151, Mar. 2022, pp. 2372–2396.

[17] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra, "PathNet: Evolution channels gradient descent in super neural networks," 2017, *arXiv:1701.08734*.

[18] A. Mallya and S. Lazebnik, "PackNet: Adding multiple tasks to a single network by iterative pruning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7765–7773.

[19] C. D. Mustafa B Gurbuz, "NISPA: Neuro-inspired stability-plasticity adaptation for continual learning in sparse networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 162, Jul. 2022, pp. 8157–8174.

[20] A. Mallya, D. Davis, and S. Lazebnik, "Piggyback: Adapting a single network to multiple tasks by learning to mask weights," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 67–82.

[21] 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)*, Jun. 2021, pp. 3013–3022.

[22] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, and G. Tesauro, "Learning to learn without forgetting by maximizing transfer and minimizing interference," in *Proc. Int. Conf. Learn. Represent.*, May 2019, pp. 1–7.

[23] D. Shim, Z. Mai, J. Jeong, S. Sanner, H. Kim, and J. Jang, "Online class-incremental continual learning with adversarial Shapley value," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 11, 2021, pp. 9630–9638.

[24] R. Tiwari, K. Killamsetty, R. Iyer, and P. Shenoy, "GCR: Gradient coreset based replay buffer selection for continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 99–108.

[25] G. Saha and K. Roy, "Saliency guided experience packing for replay in continual learning," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2023, pp. 5262–5272.

[26] L. Caccia, R. Aljundi, N. Asadi, T. Tuytelaars, J. Pineau, and E. Belilovsky, "New insights on reducing abrupt representation change in online continual learning," 2021, *arXiv:2104.05025*.

[27] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara, "Dark experience for general continual learning: a strong, simple baseline," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, Dec. 2020, pp. 15920–15930.

[28] R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin, and L. Page-Caccia, "Online continual learning with maximal interfered retrieval," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Dec. 2019, pp. 11849–11860.

[29] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*.

[30] S. Thulasidasan, G. Chennupati, J. A. Bilmes, T. Bhattacharya, and S. Michalak, "On mixup training: Improved calibration and predictive uncertainty for deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Dec. 2019, pp. 13888–13899.

[31] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, and J. Choe, "CutMix: Regularization strategy to train strong classifiers with localizable features," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6022–6031.
[32] A. F. M. S. Uddin, M. S. Monira, W. Shin, T. Chung, and S.-H. Bae, "SaliencyMix: A saliency guided data augmentation strategy for better regularization," 2021, *arXiv:2006.01791*.
[33] V. Verma, "Manifold mixup: Better representations by interpolating hidden states," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6438–6447.
[34] S. Venkataramanan, E. Kijak, L. Amsaleg, and Y. Avrithis, "AlignMixup: Improving representations by interpolating aligned features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 19152–19161.
[35] J. Kim, W. Choo, H. Jeong, and H. O. Song, "Co-mixup: Saliency guided joint mixup with supermodular diversity," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–12.
[36] F. Pinto, H. Yang, S. N. Lim, P. Torr, and P. Dokania, "Using mixup as a regularizer can surprisingly improve accuracy & out-of-distribution robustness," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, Dec. 2022, pp. 14608–14622.
[37] J. Oh and C. Yun, "Provable benefit of mixup for finding optimal decision boundaries," in *Proc. 40th Int. Conf. Mach. Learn.*, Jul. 2023, vol. 202, pp. 26403–26450.
[38] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3987–3995.
[39] R. Aljundi, P. Chakravarty, and T. Tuytelaars, "Expert gate: Lifelong learning with a network of experts," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7120–7129.
[40] A. Mikolajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *Proc. Int. Interdiscipl. PhD Workshop (IIPhDW)*, May 2018, pp. 117–122.
[41] A. Chaudhry, M. Ranzato, M. Rohrbach, and M. Elhoseiny, "Efficient lifelong learning with A-GEM," in *Proc. Int. Conf. Learn. Represent.*, May 2019, pp. 1–5.
[42] Z. Mai, R. Li, H. Kim, and S. Sanner, "Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 3584–3594.
[43] Y. Guo, B. Liu, and D. Zhao, "Online continual learning through mutual information maximization," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 8109–8126.
[44] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Tech. Rep., 2009.
[45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[46] J. S. Vitter, "Random sampling with a reservoir," *ACM Trans. Math. Softw.*, vol. 11, no. 1, pp. 37–57, Mar. 1985.
[47] Y. Gu, X. Yang, K. Wei, and C. Deng, "Not just selection, but exploration: Online class-incremental continual learning via dual view consistency," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7432–7441.
[48] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9260–9269.

WON-SEON LIM received the B.S. and M.S. degrees from Chung-Ang University, Seoul, South Korea, where he is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering. His research interests include continual learning, neural architecture search, and on-device AI.

YU ZHOU (Member, IEEE) received the B.Sc. degree in electronics and information engineering and the M.Sc. degree in circuits and systems from Xidian University, Xi'an, China, in 2009 and 2012, respectively, and the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2017. He is currently a Tenured Associate Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include computational intelligence and machine learning and intelligent information processing.

DAE-WON KIM (Member, IEEE) received the B.S. degree from Kyungpook National University, Daegu, South Korea, and the M.S. and Ph.D. degrees from Korea Advanced Institute of Science and Technology. He is currently a Professor with the School of Computer Science and Engineering, Chung-Ang University, Seoul, South Korea. Prior to joining Chung-Ang University, he was a Postdoctoral Researcher with Korea Advanced Institute of Science and Technology. His research interests include advanced data mining algorithms with innovative applications to bioinformatics, music emotion recognition, educational data mining, affective computing, and robot interaction.

JAESUNG LEE received the B.S., M.S., and Ph.D. degrees in computer science from Chung-Ang University, Seoul, Republic of Korea, in 2007, 2009, and 2013, respectively. He studies classification and feature selection, especially multilabel learning with information theory. He is currently the Head and an Associate Professor with the Department of Artificial Intelligence, Chung-Ang University. Concurrently, he is the Chief of the AI/ML Innovation Research Center, Chung-Ang University. His research interests include machine learning, multilabel learning, model selection, and neural architecture search.