arXiv:2508.08677v1 [cs.LG] 12 Aug 2025

Multi-level Collaborative Distillation Meets Global Workspace Model: A Unified Framework for OCIL

Shibin Su, Guoqiang Liang, De Cheng, Shizhou Zhang, Lingyan Ran, Yanning Zhang, *Fellow, IEEE,*

***Abstract*—Online Class-Incremental Learning (OCIL) enables models to learn continuously from non-i.i.d. data streams and samples of the data streams can be seen only once, making it more suitable for real-world scenarios compared to offline learning. However, OCIL faces two key challenges: maintaining model stability under strict memory constraints and ensuring adaptability to new tasks. Under stricter memory constraints, current replay-based methods are less effective. While ensemble methods improve adaptability (plasticity), they often struggle with stability. To overcome these challenges, we propose a novel approach that enhances ensemble learning through a Global Workspace Model (GWM)—a shared, implicit memory that guides the learning of multiple student models. The GWM is formed by fusing the parameters of all students within each training batch, capturing the historical learning trajectory and serving as a dynamic anchor for knowledge consolidation. This fused model is then redistributed periodically to the students to stabilize learning and promote cross-task consistency. In addition, we introduce a multi-level collaborative distillation mechanism. This approach enforces peer-to-peer consistency among students and preserves historical knowledge by aligning each student with the GWM. As a result, student models remain adaptable to new tasks while maintaining previously learned knowledge, striking a better balance between stability and plasticity. Extensive experiments on three standard OCIL benchmarks show that our method delivers significant performance improvement for several OCIL models across various memory budgets.**

***Index Terms*—Online Class Incremental Learning, Global Workspace, Knowledge Distillation, Plasticity and Stability Balance**

I. Introduction

Class-Incremental Learning is designed to integrate the knowledge of classes from a stream of data with an evolved distribution [1]. Depending on whether the learner has unlimited access to the current task's training data for multiple epochs, existing methods can be divided into two settings: *offline* and *online*. This paper tackles the more challenging *Online Class-Incremental Learning* (OCIL) task, where the model can use data samples for only one epoch of training [2]–[4]. While OCIL is more efficient in terms of memory and computation, the one-epoch training introduces numerous challenges [5].

To mitigate the notable performance drop of previously learned tasks, known as catastrophic forgetting (CF) [6], most existing OCIL works rely on data replay techniques [3], [4], [7]–[9]. Specifically, a memory buffer is employed to store a few samples from old tasks. Then, an input batch is drawn from the data stream and merged with a randomly selected memory batch for model training. Following this basic framework, several aspects of replay techniques have been explored. Many works focus on designing efficient strategies for memory updating [3], [10] or memory retrieval [8], [11]. Besides, some methods explored how to use data stream and memory samples more efficiently such as augmenting classifiers [12], [13] and developing new loss functions [14]–[16].

While these studies have enhanced the overall accuracy through mitigating CF, they neglect the challenge of learning new tasks. Due to the one-epoch training constraint, the OCIL model encounters under-fitting and shortcut learning, leading to biased, non-essential features and inadequate generalization [17]. To enhance plasticity, Wang et al. [18] first proposed the use of two peer learners to simultaneously learn from data, which is further augmented with a distillation chain. Although it improves plasticity, the overall stability is limited. Moreover, in practical applications, the memory size limitation tends to be stricter, coupled with an increased number of classes to master. Consequently, the quantity of memory samples per class is considerably reduced, further exacerbating the challenge of maintaining old knowledge. On the other hand, cognitive scientists have built a well-known global workspace theory (GWT) [19]–[21]. In this theory, there exists a shared, dynamic workspace in the brain, called global workspace (GW). Multiple independent modules strive to transmit information to GW via attention, while GW intermittently disseminates global data back to the students in response to task demands and external stimuli. Its inherent dynamism allows the brain to flexibly adjust its information processing strategies over time, thereby improving overall cognitive efficiency.

Inspired by the GWT theory, we propose to enhance ensemble learning by introducing a global workspace for OCIL, which serves as an implicit knowledge memory and directs the learning of student models. To construct the global workspace efficiently, we draw on theoretical foundations from optimization, parameter space and generalization. The concept of linear mode connectivity suggests that models initiated identically but trained with various SGD noise generally

Manuscript received XXX. This work was supported in part by the National Natural Science Foundation of China (No. 62376218, 62101453); in part by Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515011298); in part by the Natural Science Basic Research Program of Shaanxi Province (No. 2022JC-DW-08). (Shibin Su and Guoqiang Liang contributed equally to this work) (Corresponding author: Guoqiang Liang; De Cheng.)

Shibin Su, Shizhou Zhang, Lingyan Ran, and Yanning Zhang are with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: szzhang@nwpu.edu.cn; lran@nwpu.edu.cn; ynzhang@nwpu.edu.cn).

Guoqiang Liang is with the School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China, also with Shenzhen Research Institute of Northwestern Polytechnical University, Shenzhen 518057, China(e-mail: gqliang@nwpu.edu.cn).

De Cheng is with the School of Telecommunications Engineering, Xidian University, Xi'an 710071, China (e-mail: dcheng@xidian.edu.cn).

converge to a common loss basin [22]. These diverse models can be unified through linear interpolation to form a model that inhabits a flat, low-loss region. This leads to greater robustness against perturbations compared to sharp minima [23], thereby offering more reliable and generalized solutions. Thus, we adopt a linear combination of multiple students' parameters to construct the GWM, which can reflect the upper and lower bounds of the local region's loss. To further preserve the optimal minima over past training batches, the exponential moving average (EMA) is employed to update the GWM, effectively stabilizing the loss curvature in a manner similar to sharpness-aware minimization (SAM) [24]. This moving average mechanism enables the GWM to serve as a stable reference for long-term knowledge. Therefore, we use it to direct the student learning process. Specifically, we develop a parameter fusion mechanism, where the GWM parameters are periodically propagated back to students in a specific proportion. By doing this, we can prevent the students from deviating from the historical optimal, thus ensuring the learning of students and the construction of GWM.

Moreover, we design a multi-level collaborative distillation mechanism for overall model learning. Besides the cross-entropy (CE) loss for each student, we design a knowledge distillation (KD) loss to align the outputs of the two students. Additionally, we introduce another KD loss for synchronizing the outputs of GWM and students, thereby guiding the students towards the average trajectory. Through these losses and parameter fusion, the students' parameters are pushed towards the historical optima, mitigating the risk of parameter drift while preserving the model's plasticity. Meanwhile, this effectively flattens the loss basin around the students' solution, leading to improved stability and generalization across different tasks. The overall framework is a fundamental strategy, which can be applied to a wide range of existing OCIL approaches.

Our main contributions can be summarized as follows:

- We propose to establish a global workspace model (GWM) for OCIL, which works as an implicit knowledge memory and directs the learning of multiple students.
- We devise a multi-level collaborative distillation mechanism to enforce the consistency of students by synchronizing their predictions and preserve historical knowledge by aligning each student with the GWM.
- Extensive experiments on three popular OCIL benchmarks demonstrate the effectiveness of the proposed method, achieving new state-of-the-art performance.

The rest of this article is organized as follows. Section II gives a brief review of related work. In Section III, we introduce the proposed method. Then we present and analyze the experimental results in Section IV. Finally, Section V concludes this study and outlines directions for future research.

II. Related Work

A. Online Class Incremental Learning

In OCIL, replay-based methods have gained significant attention due to their effectiveness and simplicity [5]. A pioneering replay-based work is Experience Replay (ER), which adopted a reservoir sampling algorithm and a random updating strategy for memory management [9]. Building upon this foundation, numerous replay-based variants have been developed. Some focus on enhancing memory update and retrieval strategies [8], [11], [25]. For instance, MIR [11] retrieved memory samples that are most interfered by the incoming data batch. SSD [3] condensed stream data into more informative exemplars for efficient storage.

Meanwhile, other efforts are directed at improving model learning through architectural innovations [12], [18], [26]–[28] and novel optimization objectives [15], [25], [29]–[31]. Wang et al. [12] designed a continual bias adaptor to augment the classifier. To tackle the overfitting-underfitting dilemma, MOSE-MOE [28] introduced a stacked sub-experts model, which was optimized by multi-level supervision and reverse self-distillation. Recently, Wu et al. [4] introduced a dual-domain division multiplexer to alleviate both inter-task and intra-task bias, which intervenes confounders and multiple causal factors over frequency and spatial domains.

For the objective function, ER-ACE [15] replaced the vanilla CE loss with an asymmetric variant to mitigate representation drift. OCM [14] maximized mutual information between the old and new representations. [29] introduced a gradient self-adaptive loss to solve the cross-task discrimination problem. UER [32] decomposed the conventional logits of the dot product into an angle factor and a norm factor, using the angle component to learn current samples and both components for replay samples. Based on the principle of maximum a posterior estimation, Michel et al. [33] devised a novel loss function, enforcing the learned representations to distribute on the unit sphere. Pareto optimization has also been adopted to capture the interrelationship among previously learned tasks [34]. Zhou et al. [7] proposed a balanced destruction-reconstruction module, which tries to reduce the degree of maximal destruction of old knowledge to achieve better knowledge reconstruction. Seo et al. [31] proposed preparatory data training to induce neural collapse and a residual correction module to reduce discrepancies during inference. To address task recency bias in the combination of the fully connected layer and softmax, supervised contrast learning [35] has been incorporated into OCIL and has enhanced performance [13], [36], [37]. For example, SCR [36] combined the supervised contrastive loss and nearest class mean (NCM) classifier. PCR [13] further introduced a proxy-based contrastive loss to address the imbalance issue.

B. Knowledge Distillation in OCIL

To alleviate CF, KD [38] has been widely used in offline CL, where the model learned on old tasks serves as a fixed teacher [39]–[43]. Conversely, its application in OCIL is still relatively restricted [44], [45]. Because of the only one-epoch training constraint, the model tends to experience issues such as under-fitting and shortcut learning [4], [17], leading to a less effective teacher. As it fails to capture the critical features, KD ultimately becomes more harmful than beneficial for model update [30]. To address this problem, momentum knowledge distillation was applied to OCIL in [30]. Instead of a student

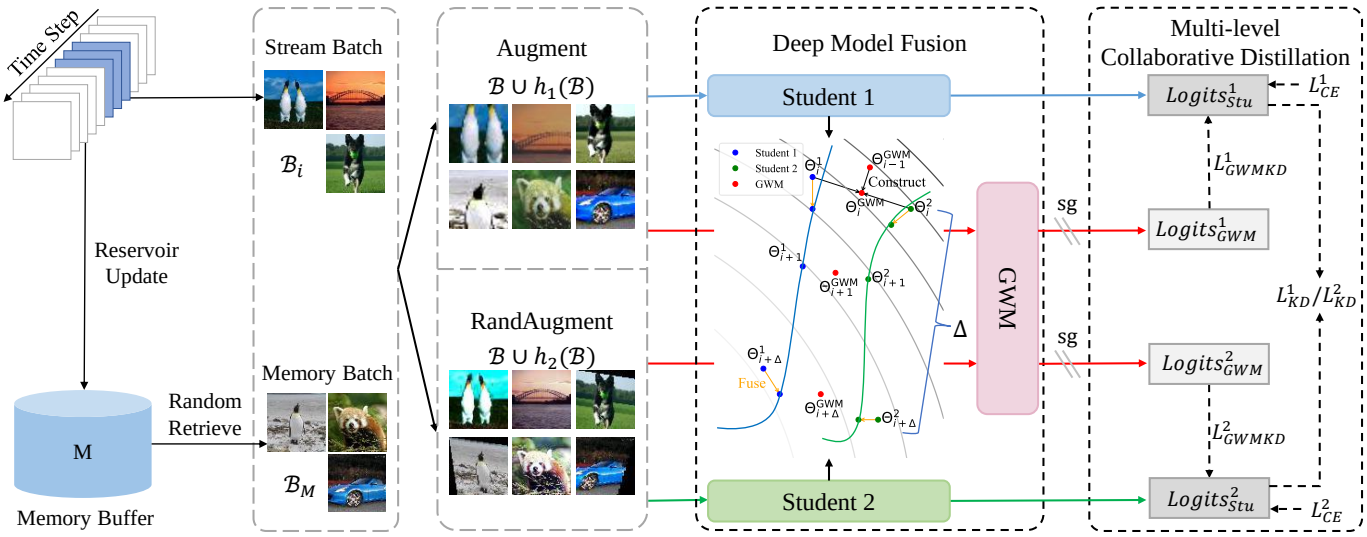


Fig. 1: Overview of the proposed method. For the combination of a stream data batch and a random retrieved memory batch, we apply two distinct augmentation strategies. The resulting augmented batches and their original batch are fed into two students and Global Workspace Model (GWM) to produce logits. In each training iteration, we use a linear combination of students' parameters to construct the GWM. Moreover, GWM's parameters are periodically fused back into students. We use blue, dark green and red lines to represent the forward processes of various models. "sg" denotes the stop-gradient operation.

and a well-trained teacher, Wang et al. [18] applied KD to two peer learners. Although enhancing model plasticity, it lacked an explicit mechanism for mitigating forgetting. In [28], reverse self-distillation was developed to gather knowledge of various experts from shallow to deep.

Unlike previous research, we focus on addressing the stability challenges of ensemble models without compromising their adaptability. To accomplish this, we develop a GWM and employ multi-level collaborative distillation to steer the learning of various student models.

III. Methodology

A. Problem Definition

An OCIL model is trained on a non-i.i.d. data stream consisting of T tasks. In the t-th task, its training data is $D_t = \{\mathcal{X}_n^t, y_n^t\}_{n=1}^{N_t}$, where $\mathcal{X}_n^t$ denotes the n-th image with label $y_n^t \in \mathcal{Y}^t$, and N_t is the number of all images in this task. The labels of any two tasks are non-overlapping, that is, $\mathcal{Y}^i \cap \mathcal{Y}^j = \emptyset$ for $\forall\ i, j \in \{1, \ldots, T\}, i \neq j$. Each sample can be used to train the model only once, unless stored in the fixed-size memory buffer $\mathcal{M}$. During testing, the model is assessed on a test set containing samples of all learned classes.

B. Overall Framework

In this paper, we propose a multi-level collaborative distillation based deep model fusion for OCIL, aiming to explicitly improve model stability while maintaining plasticity. Fig. 1 illustrates the framework of our method, which consists of two modules: data augmentation and deep model fusion. In data augmentation, each data batch from the current stream is combined with a batch retrieved from the memory buffer, resulting in a batch $\mathcal{B}$. Then, two kinds of data augmentation strategies are applied to the batch $\mathcal{B}$. Next, both the original and the different augmented batches are fed into the deep model fusion module. There are two student networks and a Global Workspace Model (GWM). The two students have identical network structure and the same random initialization. The GWM is a deep fusion of the two students, encoding the historical parameter optimization trajectory as an implicit knowledge memory. By periodically feeding the GWM back to the students, the GWM acts as a dynamic anchor to guide student learning.

To optimize our model, we devise a multi-level collaborative distillation mechanism. Besides the traditional CE loss, an online KD loss is employed between the two students. By aligning their predictions for distinct augmented versions of identical data, we can enhance the diversity of the two students, thereby improving their plasticity. In addition, we develop another KD loss for the GWM and the two students, which compels the students to adhere to historical consensus and strengthens their long-term memory.

Since there are no constraints on the student models, many replay-based CIL methods are eligible to serve as the student models. In other words, our method can be integrated with many existing replay-based models. In the following, we primarily detail deep model fusion and multi-level collaborative distillation.

C. Deep Model Fusion

The one epoch training constraint poses more challenges for optimizing the OCIL model, such as under-fitting, shortcut learning, and the risk of a narrow loss minimum. In other words, the non-essential features and the local minimum will lead to poor generalization. Moreover, due to relatively low robustness of model parameters and imbalanced quantity of

samples between old tasks and new task, the model update for new tasks will destroy the knowledge of old tasks.

To boost the convergence of online continual learners, [18] combined the ensemble model with online knowledge distillation (OKD). However, previous studies have shown that over-parameterized students tend to diverge from each other during training, even under the supervision of KD loss [46], [47]. Although OKD significantly improves the plasticity of model under one-epoch training, this loose coupling may cause the parameter space to gradually disperse, leading to the loss of a unified representation of old knowledge. Besides, the distillation relies only on the knowledge sharing between two models from the current batch, without explicitly solidifying old knowledge. Both of these aspects severely affect the model stability. Moreover, the plasticity in [18] only represents the learning performance of the model on the current task. It is not sure whether the features learned by the OKD model really generalize, and whether the generalization of the model across tasks really improves. Hence, explicitly enhancing model stability while maintaining plasticity is very important for ensemble models.

Instead of aligning the outputs of various models, we propose to directly manipulate model parameters. Specifically, we construct a global workspace based on parameter fusion to enhance the stability and generalization of the model while maintaining plasticity. Firstly, we construct a GWM by performing a linear combination of the parameters of multiple students, which approximates the solution of the parameter space surrounding the student models.

$$\Theta_i^{\mathrm{GWM}} = \sum_{m=1}^{M} r_i^m \Theta_i^m, \quad \mathbf{r}_i = [r_i^1, \ldots, r_i^M] \sim \mathrm{Dir}(\xi), \quad (1)$$

where i denotes the i-th training batch, Θ_i^{GWM} and Θ_i^m are the parameters of feature extractor for GWM and the m-th student in the i-th training batch, M is the number of students, and $\mathbf{r}_i$ is a weight vector constrained by $\sum_{m=1}^{M} r_i^m = 1$. The distribution $\mathrm{Dir}(\xi)$ refers to the Dirichlet distribution parameterized by $\xi \in \mathcal{R}^M$. In our actual experimental setting, there are two students. And we set r^1 and r^2 as 0.5 for all iterations. Since the classifier weight $\mathrm{W}_i^{\mathrm{GWM}}$ and $\mathrm{W}_i^{\mathrm{m}}$ are updated similarly, we just detail the formulation for the feature extractor for simplicity. As the GWM represents the surrounding parameter space encompassed by multiple students, its classification loss showcases the maximum and minimum bounds of loss in the vicinity of the student models [48]. Therefore, minimizing this loss can flatten the loss surface and establish a more stable loss minimum. This pursuit of flatter minima serves as an implicit form of regularization, improving model generalization and enabling the student models to converge to a wider and more robust solution space.

In Eq. (1), the GWM model is updated at each training batch, thus obtaining optimal parameters for the current data batch. However, due to the one-epoch training and extremely limited memory batch size, sample noise and outliers will induce parameter fluctuations, thereby impacting the training process. To establish a stable reference and retain knowledge over more data batches, we integrate the parameters of the current batch with those from the previous batches using the exponential moving average:

$$\Theta_i^{\mathrm{GWM}} = (1-\alpha)\Theta_{i-1}^{\mathrm{GWM}} + \alpha\Theta_i^{\mathrm{GWM}}, \quad (2)$$

where α is a hyper-parameter. This fusion not only facilitates a smoother and more stable training procedure but also enables the GWM model to act as a bridge between new and old tasks.

The two students are trained using distinct augmented data batches, leading their parameters to diverge significantly. Thus, directly taking their average will produce a poor model because of the non-linear nature of deep neural network. To mitigate excessive divergence, we design a parameter fusion mechanism to explicitly force the student parameters towards the history-smoothing space path. In particular, we fuse the GWM and students in a specific proportion at regular intervals:

$$\text{If } \mathrm{mod}(i, \Delta) = 0 : \Theta_i^m = (1-\gamma)\Theta_i^m + \gamma\Theta_i^{\mathrm{GWM}}, \quad (3)$$

where Δ and γ are two hyper-parameters denoting the fusion interval and ratio, respectively. Δ can be set at the task or batch level, such as every task or every fifty batches. According to Eq. (3), the parameters of students are confined to the vicinity surrounding the GWM. As a result, the students not only learn different patterns but also avoid excessive divergence.

D. Multi-level Collaborative Distillation

The architecture of our model consists of multiple students and a GWM. To promote effective interaction, we design a multi-level collaborative distillation mechanism for overall model learning, which comprises the CE loss, KD loss between students and KD loss between students and GWM.

For each student, we first utilize the ground-truth labels of samples to steer its learning process. In particular, if $f(\cdot;\Theta^1)$ represents the feature extractor of student 1, the CE loss of student 1 is computed as follows:

$$L_{\mathrm{CE}}^1 = -\sum_{(\mathcal{X},y)\in\mathcal{B}\cup h_1(\mathcal{B})} \log\left(\frac{e^{f(\mathcal{X};\Theta^1)\cdot\mathbf{w}_y^1}}{\sum_{j\in\mathbf{C}_t} e^{f(\mathcal{X};\Theta^1)\cdot\mathbf{w}_j^1}}\right), \quad (4)$$

where $\mathbf{C}_t$ signifies the set of all observed classes up to task t, h_1 denotes the data augmentation for student 1, y is the ground-truth label of sample $\mathcal{X}$, and $\mathbf{w}_y^1$ represents the weight for class y in the classifier of student 1.

While both students receive distinct augmented data from the same data batch, it is infeasible for them to interact solely through the CE loss. To enhance cross-student interaction, we develop a KD loss to align the predicted probability from the two students. For a sample $\mathcal{X}$, the probability that the student 1 assigns it to class $c \in \mathbf{C}_t$ is given by

$$p_c^1 = \frac{e^{f(\mathcal{X};\Theta^1)\cdot\mathbf{w}_c^1/\tau}}{\sum_{j\in\mathbf{C}_t} e^{f(\mathcal{X};\Theta^1)\cdot\mathbf{w}_j^1/\tau}}, \quad (5)$$

where τ is the temperature hyper-parameter. Therefore, $\mathbf{p}^1(\mathcal{X}) = \{p_c^1 : c \in \mathbf{C}_t\}$ symbolizes the predicted probabilities for all seen classes according to student 1. $\mathbf{p}^2(\mathcal{X})$ is derived

similarly for student 2. Given these probabilities, the KD loss between students can be expressed as

$$L_{\text{KD}}^1 = \sum_{(\mathcal{X},y)\in\mathcal{B}} D\left(\mathbf{p}^1(\mathcal{X})||\mathbf{p}^2(\mathcal{X})\right) + D\left(\mathbf{p}^1(h_1(\mathcal{X}))||\mathbf{p}^2(h_2(\mathcal{X}))\right), \quad (6)$$

where $D(\cdot \parallel \cdot)$ is the KL divergence. In Eq. (6), we also align the probabilities across various augmented views of the same data, improving the multi-view consistency of the students.

Furthermore, to encourage students towards average trajectory, we introduce another KD loss for outputs of GWM and students. Specifically, for student 1, it can be calculated as

$$L_{\text{GWMKD}}^1 = \sum_{(\mathcal{X},y)\in\mathcal{B}\cup h_1(\mathcal{B})} D\left(\mathbf{p}^1(\mathcal{X})||\mathbf{p}^{\text{GWM}}(\mathcal{X})\right), \quad (7)$$

where $\mathbf{p}^{\text{GWM}}(\mathcal{X})$ denotes the predicted probabilities of all observed classes from the GWM model. Utilizing the GWM as a dynamic knowledge anchor, this equation helps avoid significant deviations in students, which might cause the loss landscape to a narrow crevice. By maintaining this, we can retain the model's sensitivity to the crucial features of previous tasks, thus enhancing cross-task generalization.

Ultimately, by combining the above CE loss, KD loss and GWM KD loss, we can establish the overall multi-level collaborative distillation (MCD) loss for student 1:

$$L_{\text{MCD}}^1 = L_{\text{CE}}^1 + L_{\text{KD}}^1 + \lambda L_{\text{GWMKD}}^1, \quad (8)$$

where λ is a hyper-parameter regulating the proportion of alignment from the GWM model. The loss L_{MCD}^2 for student 2 can be calculated similarly.

E. Overall Optimization Objective

To optimize the model parameters, the overall training loss L^1 for student 1 incorporates its original loss together with the aforementioned MCD loss,

$$L^1 = L_{\text{Baseline}}^1 + L_{\text{MCD}}^1, \quad (9)$$

where L_{Baseline}^1 denotes the loss function of an existing model to which our model adapts. L^2 is computed for student 2 in a similar manner.

During inference, a test sample is fed into two student models to produce their respective probabilities. We take their average as the final prediction. The entire process of training and inference is detailed in Algorithm 1.

Algorithm 1 Multi-Level Collaborative Distillation based Deep Model Fusion for OCIL

Input: Memory buffer size M_s; Learning rate lr
Init: Memory buffer $\mathcal{M} \leftarrow \{\} * M_s$; Number of observed samples $n \leftarrow 0$; Parameters of student 1 $\{\Theta^1,\mathbf{W}^1\}$, student 2 $\{\Theta^2,\mathbf{W}^2\}$ and GWM $\{\Theta^{\text{GWM}}, \mathbf{W}^{\text{GWM}}\}$; Student 1 and Student 2 are initialized identically; Two AdamW optimizers $optim1$ and $optim2$.

for $t \in \{1,\ldots,T\}$ **do**
  // Training Phase
  for $\mathcal{B}_i \sim \mathcal{D}_t$ **do**
    $\mathcal{B}_M \leftarrow RandomRetrieve(\mathcal{M})$ // Memory batch
    $\mathcal{B} \leftarrow \mathcal{B}_i \cup \mathcal{B}_M$
    // Perform different data augmentation
    Do data augmentation $\overline{\mathcal{B}}_1 \leftarrow \mathcal{B} \cup h_1(\mathcal{B})$
    Do RandAugment $\overline{\mathcal{B}}_2 \leftarrow \mathcal{B} \cup h_2(\mathcal{B})$
    Calculate the feature representation
    Calculate probability prediction via Eq. (5)
    // Update student 1 parameters. Student 2 is similar
    Calculate the CE loss L_{CE}^1 via Eq. (4)
    Calculate the KD loss L_{KD}^1 via Eq. (6)
    Calculate the GWM KD loss L_{GWMKD}^1 via Eq. (7)
    Calculate L_{MCD}^1 by $L_{\text{MCD}}^1 \leftarrow L_{\text{CE}}^1 + L_{\text{KD}}^1 + \lambda L_{\text{GWMKD}}^1$
    Calculate L^1 by $L^1 \leftarrow L_{\text{Baseline}}^1 + L_{\text{MCD}}^1$
    $\Theta^1, \mathbf{W}^1 \leftarrow optim1(L^1, \Theta^1, \mathbf{W}^1, lr)$
    // Update GWM parameters
    Update Θ^{GWM}, $\mathbf{W}^{\text{GWM}}$ for GWM via Eq. (1 & 2)
    // Parameter fusion between GWM and Students
    if $(i+1) \mod \Delta == 0$ **then**
      Update student 1 and 2 using GWM via Eq. (3)
    end if
    // Memory Update
    $\mathcal{M} \leftarrow ReservoirUpdate(\mathcal{M}, \mathcal{B}_i, M_s, n)$
    $n \leftarrow n + |\mathcal{B}_i|$
  end for
  // Inference Phase (Optional)
  for $\mathcal{X} \sim \mathcal{D}_{test}$ **do**
    Compute probability for $c \in \mathbf{C}_t = \cup_{k=1}^t \mathcal{Y}_k$
    Predict the label by $y' \leftarrow \arg\max_{c\in\mathbf{C}_t} \frac{\mathbf{p}_c^1(\mathcal{X})+\mathbf{p}_c^1(\mathcal{X})}{2}$
  end for
end for

IV. Experiments

A. Experiment Setup

1) Evaluation Datasets and Metrics: Following previous work [3], [18], we conduct experiments on three widely used datasets for OCIL: split CIFAR-100 [49], split Tiny-ImageNet [50] and split ImageNet-100 [51]. Both CIFAR-100 and ImageNet-100 are divided into 10 tasks with 10 classes per task. The split Tiny-ImageNet contains 100 disjoint tasks, each of which includes 2 classes.

Following prior research [18], we typically present the final average accuracy (FAA) and final relative forgetting (FRF) to assess performance across all tasks, which are defined as:

$$A_T = \frac{1}{T}\sum_{j=1}^{T} a_{T,j}, \quad (10)$$

$$RF_T = \frac{1}{T}\sum_{j=1}^{T} f_{T,j},$$

$$\text{s.t. } f_{T,j} = \max_{l\in\{1,\ldots,T\}} \left(\frac{a_{l,j} - a_{T,j}}{a_{l,j}}\right), \quad (11)$$

where $a_{T,j}$ and $f_{T,j}$ denote the accuracy and relative forgetting rate of task j after training model on the last task T, respectively. Relative forgetting measures how much proportion of

TABLE I: Comparison of FAA (%) with constrained memory sizes for individual baselines, those integrated with CCL, and those combined with our approach. The best scores are highlighted in **boldface**. All results are the average of 5 runs.

Dataset	CIFAR-100 (10 tasks)			Tiny-ImageNet (100 tasks)			ImageNet-100 (10 tasks)		
Memory Size (M_s)	0.1K	0.2K	0.5K	0.2K	0.5K	1K	0.2K	0.5K	1K
ER (2019) [9]	7.4±0.7	8.6±0.5	10.9±1.1	0.9±0.1	0.9±0.1	1.0±0.1	8.1±1.6	11.6±1.6	14.9±0.8
ER+CCL-DC (CVPR 2024) [18]	11.8±1.1	15.1±1.2	23.3±1.4	3.1±0.5	6.8±0.4	6.8±1.8	11.8± 1.3	17.5± 1.1	24.3± 0.5
ER+Ours	**21.6 ± 1.4**	**28.0 ± 0.9**	**34.5 ± 0.8**	**7.6 ± 0.5**	**11.5 ± 0.7**	**15.6 ± 0.8**	**14.8 ± 1.1**	**21.9 ± 0.3**	**28.9 ± 0.7**
SCR (CVPR 2021) [36]	8.1±0.9	11.0±1.1	15.3±1.3	2.8± 0.7	6.2± 0.3	8.6± 0.6	9.2± 0.9	14.3± 0.5	16.9± 0.9
SCR+CCL-DC (CVPR 2024) [18]	12.0±1.3	17.8±1.3	28.7±1.2	3.1± 0.9	8.5± 0.6	13.0± 1.0	13.2± 1.1	22.4± 1.7	**31.9 ± 1.5**
SCR+Ours	**12.4 ± 1.3**	**19.2 ± 1.4**	**29.0 ± 0.5**	**3.7 ± 1.4**	**9.8 ± 1.1**	**14.6 ± 0.8**	**13.9 ± 1.6**	**22.5 ± 1.4**	31.4± 1.2
ER-ACE (ICLR 2022) [15]	10.4± 1.4	14.1± 2.5	19.0± 2.4	3.9± 0.6	5.9± 0.3	8.6± 0.5	13.6± 0.8	18.2± 2.0	22.8± 1.4
ER-ACE+CCL-DC (CVPR 2024) [18]	15.2± 0.7	19.6± 0.6	27.4± 0.8	5.8± 0.6	8.6± 0.7	11.5± 0.8	18.1± 1.5	26.7± 0.4	32.5± 1.8
ER-ACE+Ours	**17.1 ± 1.3**	**23.1 ± 1.3**	**29.9 ± 2.3**	**6.3 ± 0.5**	**9.2 ± 0.5**	**12.6 ± 1.0**	**22.3 ± 1.3**	**29.3 ± 1.6**	**35.6 ± 0.8**
OCM (ICML 2022) [14]	7.2±0.3	10.1±0.9	15.5±1.1	4.5±0.7	7.5±0.4	10.1±0.6	7.2±1.1	9.2±1.2	12.2±0.7
OCM+CCL-DC (CVPR 2024) [18]	9.4±1.0	11.9±0.7	17.4±1.6	4.7±0.6	7.9±0.9	12.9±1.1	10.4±0.4	13.8±1.1	19.7±1.8
OCM+Ours	**14.9 ± 0.8**	**19.9 ± 0.6**	**27.4 ± 0.4**	**8.1 ± 1.0**	**11.9 ± 1.5**	**16.1 ± 0.5**	**14.4 ± 3.4**	**21.3 ± 1.9**	**28.3 ± 2.3**
GSA (CVPR 2023) [29]	12.1± 0.6	14.6± 0.9	19.8± 1.8	3.5± 0.8	5.2± 0.6	7.1± 0.6	11.5± 0.8	15.7± 1.6	20.2± 1.1
GSA+CCL-DC (CVPR 2024) [18]	12.8± 1.2	16.5± 0.6	25.3± 0.9	2.2± 0.5	3.8± 1.0	7.8± 2.2	12.1± 0.9	17.8± 1.4	25.1± 1.0
GSA+Ours	**18.8 ± 1.3**	**24.8 ± 1.3**	**32.0 ± 1.5**	**7.8 ± 0.5**	**11.2 ± 0.6**	**14.6 ± 1.2**	**16.1 ± 1.2**	**23.3 ± 1.4**	**30.2 ± 3.2**
PCR (CVPR 2023) [13]	15.1± 0.7	19.0± 0.7	25.7± 1.7	5.8± 0.8	8.3± 0.6	11.6± 0.9	16.3± 1.9	22.1± 1.2	27.2± 1.2
PCR+CCL-DC (CVPR 2024) [18]	14.7± 1.4	19.3± 1.2	25.5± 1.5	3.1± 1.2	8.0± 1.0	12.6± 1.1	12.5± 1.3	18.8± 1.5	27.1± 3.2
PCR+Ours	**26.8 ± 2.2**	**31.3 ± 0.5**	**36.3 ± 0.9**	**8.7 ± 1.0**	**13.2 ± 0.7**	**16.2 ± 0.5**	**22.6 ± 0.7**	**30.9 ± 0.4**	**36.5 ± 1.2**
MLG (PR 2025) [52]	15.6± 1.1	18.6± 1.1	24.4± 0.8	3.4± 1.1	6.0± 0.8	8.5± 1.2	17.2± 1.6	22.8± 0.8	27.6± 1.0
MLG+CCL-DC (CVPR 2024) [18]	16.5± 1.2	20.9± 1.0	29.0± 1.0	5.6± 0.9	9.8± 0.8	13.3± 1.2	20.9± 1.8	26.7± 2.0	31.8± 1.3
MLG+Ours	**22.7 ± 0.8**	**28.3 ± 1.2**	**34.6 ± 0.5**	**7.5 ± 0.6**	**11.2 ± 0.7**	**15.0 ± 1.1**	**22.8 ± 2.5**	**30.9 ± 1.1**	**35.4 ± 0.7**
MOSE-MOE (CVPR 2024) [28]	14.7± 0.7	19.4± 1.2	27.2± 0.9	4.7± 0.5	9.1± 0.4	13.2± 1.3	16.7± 1.8	26.0± 1.1	31.3± 2.0
MOSE-MOE+CCL-DC (CVPR 2024) [18]	14.4± 1.1	19.5± 2.2	31.7± 1.1	5.1±0.9	9.5±1.2	16.8±0.6	15.8±2.1	24.8±2.1	36.0±3.1
MOSE-MOE+Ours	**20.0 ± 1.5**	**27.0 ± 1.9**	**36.2 ± 1.1**	**8.7 ± 0.6**	**14.4 ± 0.7**	**19.3 ± 0.4**	**18.9 ± 2.4**	**29.8 ± 1.4**	**39.3 ± 1.2**

performance the model forgets. In contrast to traditional forgetting measure, relative forgetting alleviates the bias towards poor plasticity, providing a fairer evaluation. Additionally, we use average learning accuracy (ALA) LA_T to evaluate the model's plasticity, adhering to the definition in [18].

$$LA_T = \frac{1}{T}\sum_{j=1}^{T} a_{j,j}. \tag{12}$$

A superior performance is indicated by a higher FAA or ALA, or a lower FRF. All reported results are the average of 5 runs to reduce randomness. In each experimental run, the order of classes is shuffled for all datasets before splitting the tasks.

To assess our method under both restricted and standard memory sizes, we perform experiments utilizing extremely limited memory as well as standard memory capacities. Under constrained conditions, the memory size is configured to 0.1K, 0.2K, 0.5K for CIFAR-100, and 0.2K, 0.5K, 1K for Tiny-ImageNet and ImageNet-100. Under standard conditions, the memory size is increased ten times.

2) Implementation Details: We employ the full ResNet-18 network without pre-training as the backbone for all methods and datasets. Like [9], random retrieval and reservoir sampling are used for memory management, which are denoted as $RandomRetrieve(\cdot)$ and $ReservoirUpdate(\cdot)$ in Algorithm 1. As we focus on extremely constrained memory and small memory batch size, the batch size is set to 10 for both the data stream and memory samples. To ensure a fair comparison, the same optimizer AdamW is applied for all experiments. Note that two separate AdamW optimizers are utilized for the two student models in our model. We tune the learning rate, weight decay, and other hyper-parameters for each baseline by maximizing its FAA score. Then, they are maintained when applying CCL-DC and our approach. Refer to the YAML files in our code repository for their detailed values. The values of γ, Δ and α are assigned to 0.5, 1 Task and 0.01, respectively. The temperature hyper-parameters τ in L_{KD} and L_{GWMKD} are configured at 4.0. The optimal value of λ depends on the baseline, so refer to the released code for its exact configuration.

Different augmentation techniques are applied to two student models. For student 1, we apply a transformation operation including random crop, random horizontal flip, color jitter and random grayscale. In contrast, RandAugment [53] is used for student 2. It contains two extra hyper-parameters N and M, which are set as 3 and 15 in all experiments. Besides, several baselines possess their own data augmentation strategies. For a fair comparison, both models trained with CCL-DC and our method maintain these unique data augmentation techniques. Specifically, the ER, SCR, and PCR utilize random cropping, horizontal flipping, color jitter and random grayscale. The color jitter parameters are set to (0.4, 0.4, 0.4, 0.1) with a probability of 0.8, while the random grayscale is applied with a probability of 0.2. For ER-ACE, OCM, GSA and MOSE-MOE, the augmentation consists of random cropping and random horizontal flip. Notably, OCM introduces global rotation augmentation combined with inner flipping, generating 15 times more training samples. GSA and MOSE-MOE adopt the inner flip operation to double the training samples.

B. Comparison with SOTA Methods

To evaluate the effectiveness of the proposed method, we applied it to several typical OCIL methods, including ER [9], SCR [36], ER-ACE [15], OCM [14], GSA [29], PCR [13], MLG [52], and MOE-MOSE [28]. For comparative analysis,

TABLE II: Comparison of FAA (%) with standard memory sizes for individual baselines, those integrated with CCL, and those combined with our approach. The best scores are indicated in **boldface**. All results are the average of 5 runs.

Dataset	CIFAR-100 (10 tasks)			Tiny-ImageNet (100 tasks)			ImageNet-100 (10 tasks)		
Memory Size (M_s)	1K	2K	5K	2K	5K	10K	2K	5K	10K
ER (2019) [9]	14.3±1.0	20.3±0.8	29.2±1.6	0.9±0.2	1.3±0.3	1.5±0.4	14.6±2.0	20.9±2.2	24.5±2.0
ER+CCL-DC (CVPR 2024) [18]	27.9±1.1	30.6±2.0	31.2±1.4	7.2±1.7	7.7±0.5	8.3±1.9	32.1± 1.7	37.4± 1.1	40.1± 0.4
ER+Ours	**38.4 ± 0.9**	**42.0 ± 1.2**	**43.7 ± 1.6**	**19.1 ± 0.9**	**19.5 ± 1.9**	**18.8 ± 1.5**	**32.9 ± 2.3**	**40.6 ± 1.0**	**42.4 ± 1.3**
SCR (CVPR 2021) [36]	18.4±0.8	20.3±0.9	22.4±0.4	10.8± 0.8	12.7± 0.7	13.8± 0.8	19.0± 1.4	20.4± 1.1	20.7± 1.1
SCR+CCL-DC (CVPR 2024) [18]	35.3±0.8	38.7±1.0	40.3±0.6	15.0± 0.9	15.9± 1.2	17.1± 1.2	39.5± 1.3	44.4± 0.9	45.1± 1.6
SCR+Ours	**37.0 ± 1.0**	**41.2 ± 1.1**	**43.2 ± 1.2**	**20.5 ± 0.8**	**23.1 ± 0.6**	**23.9 ± 0.8**	**39.5 ± 0.9**	**44.9 ± 0.7**	**46.5 ± 1.3**
ER-ACE (ICLR 2022) [15]	22.3± 2.3	24.5± 0.9	26.1± 1.4	11.4± 0.5	11.7± 0.9	10.9± 1.1	25.9± 1.6	29.8± 1.0	32.9± 1.0
ER-ACE+CCL-DC (CVPR 2024) [18]	31.3± 1.0	34.7± 1.2	35.4± 1.0	14.8± 1.1	16.6± 1.9	17.2± 2.0	38.0± 1.6	43.0± 0.7	43.5± 2.7
ER-ACE+Ours	**34.0 ± 0.5**	**37.6 ± 1.9**	**39.9 ± 1.3**	**15.9 ± 1.1**	**19.2 ± 2.0**	**19.6 ± 2.1**	**39.8 ± 0.7**	**44.0 ± 1.1**	**46.1 ± 1.1**
OCM (ICML 2022) [14]	16.8±1.6	17.6±2.9	19.3±1.8	12.9±0.8	13.3±1.6	14.5±1.4	14.2±1.6	14.6±2.4	14.9±2.0
OCM+CCL-DC (CVPR 2024) [18]	20.2±2.7	21.3±1.5	20.5±2.7	17.5±1.1	19.2±3.0	21.1±0.6	24.9±1.5	30.2±2.6	32.6±2.3
OCM+Ours	**33.7 ± 1.2**	**36.0 ± 0.8**	**36.5 ± 1.0**	**21.3 ± 1.4**	**23.9 ± 1.1**	**23.9 ± 2.0**	**34.4 ± 1.8**	**42.1 ± 1.2**	**45.1 ± 1.6**
GSA (CVPR 2023) [29]	23.8± 0.9	26.2± 2.2	27.9± 1.9	9.3± 1.6	12.6± 0.8	13.7± 1.5	25.7± 1.9	32.8± 2.1	35.9± 1.2
GSA+CCL-DC (CVPR 2024) [18]	31.5± 0.9	37.8± 1.1	41.5± 2.5	13.5± 1.4	18.8± 1.4	18.8± 1.1	32.7± 1.6	43.5± 0.7	48.4± 1.8
GSA+Ours	**37.7 ± 0.9**	**42.7 ± 1.4**	**45.8 ± 0.8**	**17.9 ± 1.2**	**21.0 ± 0.9**	**24.5 ± 1.3**	**38.4 ± 1.3**	**45.9 ± 1.4**	**49.0 ± 1.3**
PCR (CVPR 2023) [13]	29.3± 1.1	31.7± 1.2	33.6± 0.9	12.8± 0.8	15.0± 1.3	15.2± 1.2	32.7± 1.5	36.9± 2.8	38.8± 1.8
PCR+CCL-DC (CVPR 2024) [18]	30.8± 1.5	34.1± 1.9	36.1± 1.8	13.7± 1.3	15.6± 1.5	16.3± 1.0	34.1± 2.0	39.6± 3.4	42.6± 2.7
PCR+Ours	**40.0 ± 1.1**	**42.0 ± 0.9**	**43.7 ± 0.8**	**19.0 ± 0.9**	**21.2 ± 0.6**	**22.4 ± 0.7**	**41.7 ± 0.9**	**45.5 ± 0.8**	**46.4 ± 1.7**
MLG (PR 2025) [52]	28.5± 1.0	31.3± 0.6	33.6± 0.8	10.3± 0.7	11.7± 0.6	11.9± 0.8	31.0± 0.8	33.3± 1.6	34.6± 0.8
MLG+CCL-DC (CVPR 2024) [18]	32.5± 1.2	35.2± 1.0	37.2± 0.4	15.9± 1.1	18.4± 0.9	18.9± 1.2	34.8± 1.2	38.1± 1.0	39.2± 1.7
MLG+Ours	**38.4 ± 1.1**	**40.8 ± 0.8**	**43.5 ± 0.9**	**21.0 ± 1.6**	**25.1 ± 0.5**	**26.4 ± 0.5**	**40.0 ± 1.0**	**43.5 ± 1.0**	**44.5 ± 0.9**
MOSE-MOE (CVPR 2024) [28]	33.3± 0.8	38.7± 0.7	41.5± 0.8	17.3± 1.1	20.7± 1.0	20.7± 3.4	37.3± 2.9	43.3± 2.8	43.7± 2.5
MOSE-MOE+CCL-DC (CVPR 2024) [18]	40.6± 1.1	46.8± 1.1	50.3± 0.6	22.1±0.5	27.4±2.5	26.0±1.7	45.8±1.9	53.3±0.7	55.1±0.6
MOSE-MOE+Ours	**42.2 ± 1.4**	**48.4 ± 0.9**	**51.6 ± 0.4**	**24.4 ± 0.7**	**29.6 ± 0.9**	**31.2 ± 0.5**	**46.3 ± 2.1**	**53.9 ± 1.5**	**56.8 ± 0.7**

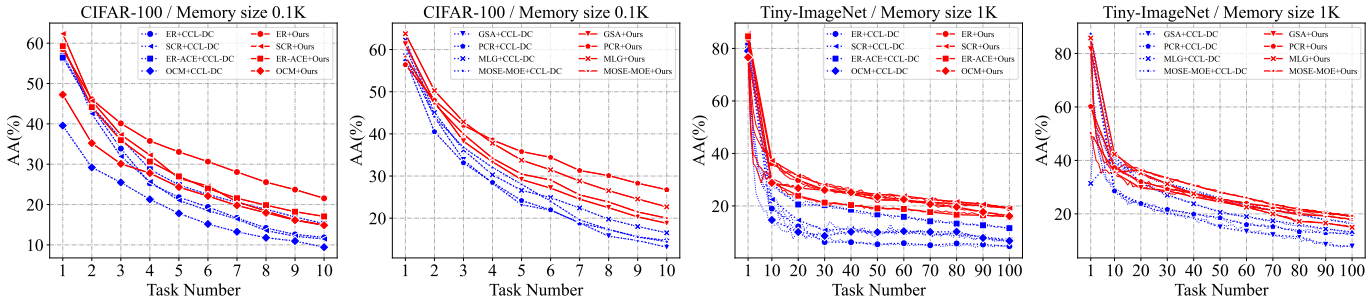


Fig. 2: The average accuracy after each incremental step for various baselines when incorporating CCL-DC and ours on CIFAR-100 and Tiny-ImageNet. Diverse markers represent distinct baselines, where the colors blue and red denote CCL-DC and Ours, respectively.

we additionally present the results of these baseline methods, along with their combination with CCL-DC [18].

Table I and II present the FAA performance of all methods across the three datasets. It can be observed that our method consistently enhances the performance of various baselines by a large margin, and surpasses the baselines+CCL-DC in nearly all cases. For example, when using ER as the baseline, we achieve an accuracy increase of $9.8\% \sim 12.9\%$ over CCL-DC on the CIFAR-100 dataset. When working with highly limited memory capacities, our approach significantly enhances FAA. Conversely, the CCL-DC method exhibits only slight improvement and occasionally reduces performance for some latest methods. Taking the most effective model, MOSE-MOE, as an example, applying our method results in increments of 5.3%, 7.6%, and 9.0% under limited memory conditions. In contrast, the CCL-DC shows performance changes of -0.3%, 0.1%, and 4.5%.

Fig. 2 displays the average accuracy of all observed tasks after each incremental step on CIFAR-100 (M_s = 0.1K) and Tiny-ImageNet (M_s=1K). The blue and red lines represent the accuracy curves after the baselines are combined with CCL-DC and our method, respectively. Obviously, for all baselines, our approach consistently attains the highest average accuracy at every step.

To evaluate the balance between stability and plasticity, we visualize the interplay between ALA and FRF on three datasets in Fig. 3. The closer a point lies to the top-left corner, the better the balance is struck. In comparison with CCL-DC, our method achieves a lower forgetting rate while maintaining comparable learning accuracy. This demonstrates that our approach effectively enhances stability without compromising plasticity, thereby achieving a more favorable balance.

C. Ablation Study

1) Effect of Proposed Modules: Firstly, we analyze the effect of key modules including the KD between students (Eq.

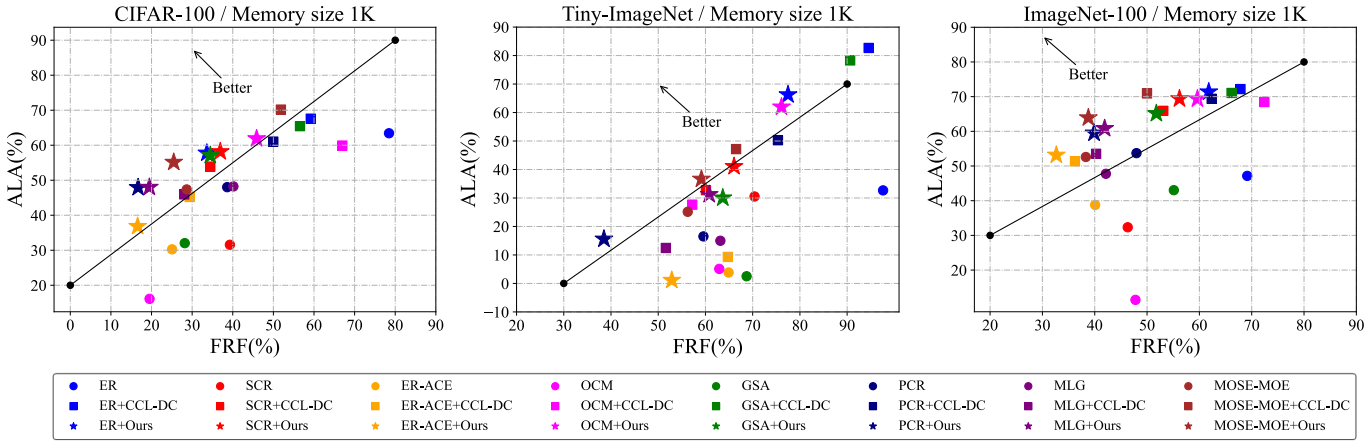


Fig. 3: Comparison of stability and plasticity balance on three datasets. The ● (circles), ■ (squares), and ★ (stars) represent the baseline, CCL+DC, and our method, respectively. Various colors are used to distinguish the baselines. The closer to the left upper corner, the better the balance is struck.

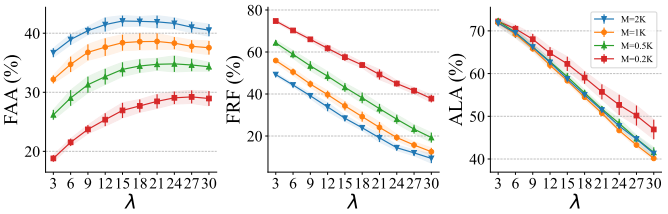
TABLE III: Effect of proposed modules using two baselines on CIFAR-100 and ImageNet-100 datasets. The terms "KD", "Fuse" and "GWMKD" denote the KD between students, parameters fusion, and KD between GWM and each student, respectively. All reported results are the average of 5 runs.

Dataset				CIFAR-100											
Memory Size(M_s)				0.1K			0.5K			1K			5K		
Baseline	KD	Fuse	GWMKD	FAA (%)	FRF (%)	ALA (%)	FAA (%)	FRF (%)	ALA (%)	FAA (%)	FRF (%)	ALA (%)	FAA (%)	FRF (%)	ALA (%)
ER				7.4±0.7	89.2±0.6	64.0±1.3	10.9±1.1	83.9±1.6	64.0±1.3	14.3±1.0	78.5±1.9	63.4±1.3	29.2±1.6	53.0±3.3	61.6±1.3
	✓			9.5±0.8	87.3±0.6	71.2±1.1	16.9±1.3	77.4±1.4	71.3±1.3	23.6±0.9	67.2±1.4	69.8±1.3	33.2±1.0	52.3±1.7	69.4±1.7
	✓	✓		11.5±0.6	85.1±0.7	74.1±1.1	21.4±1.2	71.8±1.7	73.3±1.1	27.9±0.8	62.4±1.4	72.4±0.7	36.5±1.1	50.1±1.3	72.7±1.2
	✓	✓	✓	21.6±1.4	66.1±2.1	61.0±1.5	34.5±0.8	33.9±1.8	52.1±1.6	38.4±0.9	33.6±2.0	57.8±1.4	43.7±1.6	27.0±3.9	59.9±1.8
GSA				12.1±0.6	75.2±2.1	44.8±3.8	19.8±1.8	42.8±5.6	34.8±2.4	23.8±0.9	28.2±3.0	32.0±1.4	27.9±1.9	19.6±3.4	33.1±1.4
	✓			12.3±0.8	82.7±1.9	66.1±1.8	23.4±0.8	62.8±0.9	60.4±1.3	31.5±1.1	45.4±3.2	57.0±2.1	40.0±1.8	28.5±3.9	56.1±2.0
	✓	✓		13.7±0.9	81.1±1.9	66.8±1.8	26.1±0.8	58.5±1.2	60.9±2.2	33.5±1.3	43.3±2.7	58.4±2.1	41.9±1.5	29.5±2.4	59.6±1.9
	✓	✓	✓	18.8±1.3	70.3±2.0	61.3±3.5	32.0±1.5	46.0±0.9	58.5±2.3	37.7±0.9	34.5±1.7	57.1±1.4	45.8±1.6	22.0±3.0	58.8±1.8
Dataset				ImageNet-100											
Memory Size(M_s)				0.5K			1K			5K			10K		
Baseline	KD	Fuse	GWMKD	FAA (%)	FRF (%)	ALA (%)	FAA (%)	FRF (%)	ALA (%)	FAA (%)	FRF (%)	ALA (%)	FAA (%)	FRF (%)	ALA (%)
ER				11.6±1.6	77.8±3.8	49.0±2.1	14.9±0.8	69.1±3.0	47.2±2.7	20.9±2.2	64.9±3.8	57.9±1.2	24.5±2.0	58.2±4.3	58.5±3.5
	✓			14.0±0.7	82.1±1.1	71.9±1.0	20.5±0.8	72.9±1.0	71.6±1.2	35.9±1.3	49.4±2.2	69.8±1.6	37.6±1.9	46.2±3.4	69.7±1.4
	✓	✓		17.3±0.7	78.3±1.1	75.0±0.9	24.9±0.9	67.9±0.9	74.4±1.9	39.1±0.7	47.1±0.9	73.1±1.0	41.6±1.2	43.6±0.9	73.4±1.5
	✓	✓	✓	21.9±0.3	71.0±0.7	72.2±1.1	28.9±0.7	60.5±1.2	71.4±1.1	40.6±1.0	42.8±1.3	70.8±1.7	42.4±1.3	40.4±1.1	71.4±1.7
GSA				15.7±1.6	69.0±3.2	47.3±2.2	20.2±1.1	55.1±2.3	43.0±3.0	32.8±2.1	13.6±3.5	35.7±1.5	35.9±1.2	11.0±1.1	36.4±0.7
	✓			17.0±2.0	76.9±2.1	68.6±2.1	24.8±1.8	64.8±3.1	67.2±0.9	42.6±1.1	30.2±2.6	60.5±1.5	46.1±1.8	23.5±2.5	59.9±1.7
	✓	✓		18.7±0.9	75.5±1.0	71.4±0.7	25.8±0.8	64.2±1.5	68.9±1.1	43.9±1.0	30.8±2.0	63.0±1.1	48.0±0.7	24.5±0.9	63.5±0.7
	✓	✓	✓	23.3±1.4	66.1±1.9	65.8±1.8	30.2±3.2	53.5±4.8	63.2±1.6	45.9±1.4	24.5±1.5	60.6±1.1	49.0±1.3	20.4±3.9	61.3±1.3

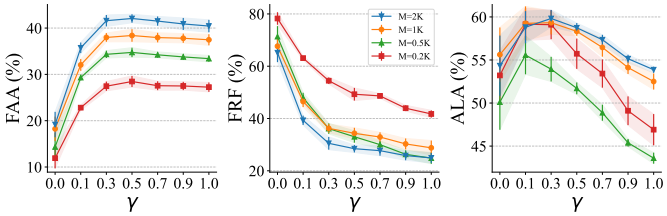
(6)), the fusion operation between GWM and students (Eq. (3)), and the KD between GWM and each student (Eq. (7)), labeled as "KD", "Fuse", and "GWMKD", respectively. Table III presents the results on CIFAR-100 and ImageNet-100, utilizing ER and GSA as baselines, respectively. Clearly, incorporating any new module can improve the FAA performance. Combining all components achieves the best performance. In particular, adding KD helps to learn new tasks, resulting in a substantial increase in ALA. When the fusion operation is further applied, ALA increases while the FRF decreases or remains stable. The fusion with GWM rectifies students' parameters, mitigating the impact of extreme noisy samples and facilitating more stable convergence on new tasks. Finally, adding GWMKD further enhances the accuracy. Moreover, when memory capacity is very limited, the performance improvement is more significant, highlighting the benefit of our approach under constrained conditions.

2) Hyper-parameter Sensitivity: To illustrate the effect of the hyper-parameters λ, γ, and Δ, we have conducted experiments utilizing ER as the baseline on CIFAR-100. We evaluated the effect of each hyper-parameter individually, keeping the other two constant.

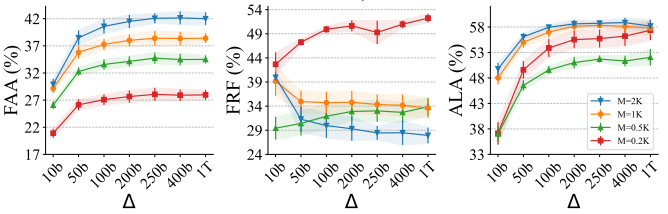
Fig. 4 (a) shows the influence of loss coefficient λ in Eq. (8). As λ increases, the FAA gradually ascends to a peak. After this point, the FAA remains relatively stable, even begins to decrease slightly. This trend remains consistent for various memory sizes. This indicates that moderate GWMKD between GWM and students is beneficial. It guides the student models towards old flatter, more generalizable solution, helping to mitigate forgetting and improving overall performance. Otherwise, excessively strong distillation causes students to blindly mimic the GWM. This overly restricts their ability to explore new tasks, ultimately resulting in a performance bottleneck. Furthermore, it is noted that for smaller memory sizes, such



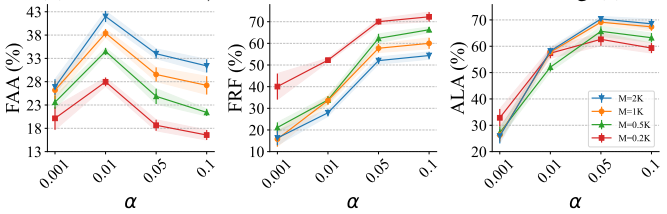
(a) Influence of λ with $\gamma = 0.5$, $\Delta = 250$.



(b) Influence of γ with $\Delta = 250$ and λ searched in Fig (a).



(c) Influence of Δ with $\gamma = 0.5$ and λ searched in Fig (a), where b and T in x-aixs are abbreviations for batch and task, respectively



(d) Influence of EMA coefficient α. According to these results, we set α as 0.01 in all experiments

Fig. 4: Impact of λ, γ, Δ, and α in ER+Ours on CIFAR-100 with varying memory sizes.

TABLE IV: FAA comparison of ER+Ours with various Δ on different datasets at 1K memory.

Datasets	10b	50b	100b	250b	500b	1000b	1Task
CIFAR-100 (500 batches per task)	29.1±0.8	35.8±1.1	37.3±1.0	38.0±1.1	**38.4 ± 0.9**	-	**38.4 ± 0.9**
Tiny-ImageNet (100 batches per task)	12.8±0.8	15.6±0.7	**15.6 ± 0.8**	-	-	-	**15.6 ± 0.8**
ImageNet-100 (1100 to 1300 batches per task)	22.3±0.9	26.2±1.6	27.2±0.9	27.9±1.1	28.5±1.1	28.8±0.7	**28.9 ± 0.7**

as 0.2K and 0.5K, a larger λ is required to compensate for the lack of replay samples. This implies that when memory samples are insufficient, stronger regularization is necessary to assist the model in achieving optimal plasticity and stability balance.

In Eq. (3), we use γ and Δ to denote the proportion of GWM and the interval of parameter fusion. Their influences are given in Fig. 4 (b) and (c), respectively. When γ equals 0, the parameters of GWM are not fused back to students (equivalent to baseline + KD + GWMKD). In other words, the two students' parameters are only restricted by aligning their probability through the two KD losses. However, the parameter trajectory of the two students will gradually diverge, leading to ineffective construction of GWM. Thus, the performance is the worst, reflecting the importance of the fusion operation between GWM and students. With the increase of γ, the FRF declines but the ALA firstly increases and then decreases rapidly. In our opinion, too strong fusion will suppress the adaptability of the model to new tasks. In a word, when γ equals 0.5, the model gets the best balance, achieving optimal FAA while maintaining lower FRF and higher ALA. Furthermore, Fig. 5 visualizes the cosine similarity of parameters between GWM and each student model, along with the similarities between the two students. When parameter fusion back is not applied ($\gamma = 0.0$), the two student models gradually diverge over time. Conversely, this divergence is effectively reduced when GWM parameters are periodically fused back into student models. For CIFAR-100 with 10 tasks, the fusion interval Δ is chosen as 1 task, resulting in 10 peaks in the $\gamma = 0.5$ curve. These results indicate the importance of our fusion operation again.

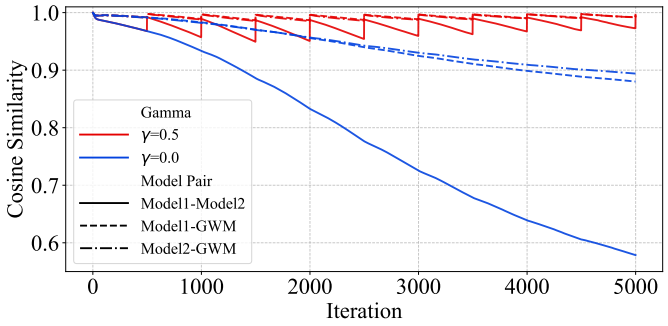


Fig. 5: Comparison of parameter cosine similarity in ER+Ours concerning parameter fusion application between GWM and students on CIFAR-100 with 1K memory. Blue lines ($\gamma = 0.0$) indicate an absence of parameter fusion while red lines ($\gamma = 0.5$) represent an average fusion between GWM and the students.

The influence of fusion interval Δ is shown in Fig. 4 (c). Too small intervals, such as 10 batches, severely restrict the model's ability to explore new tasks, resulting in a considerably reduced ALA. Raising Δ facilitates the acquisition of new tasks, but it impacts stability, particularly for much smaller memory. Due to the lack of sufficient memory samples to retain old knowledge, the increase in forgetting is more pronounced with smaller memory sizes, as illustrated by the FRF curve at 0.2K memory. When the interval Δ is greater than 250 batches, the overall FAA remains relatively stable. On the other hand, the number of images differs across datasets, leading to a varying number of batches for each task. Specifically, CIFAR-100 and Tiny-ImageNet contain 500

TABLE V: Comparison of the total running time (including both training time and inference time), GPU memory usage and FAA on CIFAR-100 at 1K memory.

	Baseline			Baseline+CCL-DC [18]			Baseline+Ours		
	Time(s)	GPU(MB)	FAA(%)	Time(s)	GPU(MB)	FAA(%)	Time(s)	GPU(MB)	FAA(%)
ER [9]	181.27	296.99	14.3	638.10	1288.31	27.9	462.87	850.59	38.4
SCR [36]	234.54	392.64	18.4	745.52	1473.45	35.3	641.48	1226.71	37.0
ER-ACE [15]	192.91	296.00	22.3	868.47	1289.06	31.3	602.29	987.37	34.0
OCM [14]	934.50	2193.49	16.8	1907.11	4688.96	20.2	1845.09	4563.26	33.7
GSA [29]	382.78	385.48	23.8	851.73	1092.97	31.5	772.58	977.76	37.7
PCR [13]	305.70	377.02	29.3	779.52	1416.69	30.8	560.54	850.38	40.0
MLG [52]	371.13	398.67	28.5	1191.41	1616.60	32.5	809.73	1352.58	38.4
MOSE-MOE [28]	533.22	534.11	33.3	2297.89	1743.49	40.6	1985.36	1494.17	42.2

TABLE VI: Comparison of FAA derived from either the predicted probability of a single student or their average on CIFAR-100.

Method	M_s	0.1K	0.2K	0.5K	1K	2K	5K
ER+Ours	Student 1	$21.6_{\pm 1.3}$	$27.7_{\pm 0.9}$	$34.3_{\pm 1.0}$	$38.3_{\pm 1.0}$	$41.8_{\pm 1.3}$	$43.6_{\pm 1.3}$
	Student 2	$21.2_{\pm 1.5}$	$27.6_{\pm 1.4}$	$34.2_{\pm 0.8}$	$38.1_{\pm 0.9}$	$41.8_{\pm 1.3}$	$43.7_{\pm 1.7}$
	Mean	**21.6** $\pm$ **1.4**	**28.0** $\pm$ **0.9**	**34.5** $\pm$ **0.8**	**38.4** $\pm$ **0.9**	**42.0** $\pm$ **1.2**	**43.7** $\pm$ **1.6**
GSA+Ours	Student 1	$18.7_{\pm 1.3}$	$24.7_{\pm 1.2}$	$31.6_{\pm 1.5}$	$37.5_{\pm 0.9}$	$42.5_{\pm 1.5}$	$45.5_{\pm 0.8}$
	Student 2	$18.8_{\pm 1.4}$	$24.7_{\pm 1.4}$	$31.9_{\pm 1.5}$	$37.6_{\pm 0.8}$	$42.5_{\pm 1.6}$	$45.7_{\pm 0.5}$
	Mean	**18.8** $\pm$ **1.3**	**24.8** $\pm$ **1.3**	**32.0** $\pm$ **1.5**	**37.7** $\pm$ **0.9**	**42.7** $\pm$ **1.4**	**45.8** $\pm$ **1.6**

and 100 batches per task respectively, while ImageNet-100 includes 1100 to 1300 batches per task. According to Table IV, we set Δ to 1 Task, obtaining the best performance across all datasets.

Furthermore, Fig. 4 (d) shows the influence of the EMA coefficient α in Eq. (2) of ER+Ours on CIFAR-100. It can be observed that the value of α is vital in achieving an optimal balance between model stability and plasticity. A smaller α results in a slower update of the GWM, inadequately capturing the present learning state. If α is too large, the GWM relies heavily on the current students, compromising stability and heightening noise sensitivity. An α value of 0.01 strikes the best balance between preserving historical knowledge and adapting to new tasks.

3) Computational Cost: Table V reports the total running time (including both training and inference) and GPU memory usage on CIFAR-100 at 1K memory. Both CCL-DC and our approach utilize dual learners, resulting in a higher computational expense than the baselines. However, compared to CCL-DC, we obtain much higher FAA accuracy with less running time and GPU memory consumption.

4) Trade-off between Efficiency and Performance: In previous sections, we merged the outputs of two students to produce the final prediction during inference. Although effective, this ensemble technique requires more computational overhead. To assess the ability of a single student, Table VI gives the FAA achieved by an individual student as well as their average. Clearly, employing just one student slightly reduces accuracy. Nevertheless, when computational resources are extremely limited, deploying just one student is a feasible choice.

5) Analysis of Feature Drift: Due to feature conflict, the adjustment of parameters for acquiring a new task will cause an excessive shift in the features for previously learned tasks.

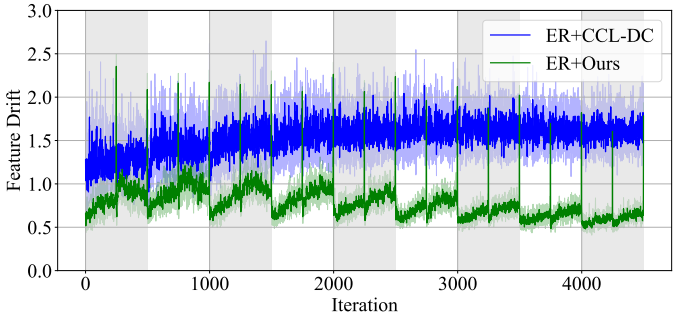


Fig. 6: Comparison of feature drift of ER+CCL-DC and ER+Ours on CIFAR-100 at 0.5K memory.

To measure this feature drift, we employ the feature distance proposed in [15], [30]. Specifically, after the i-th iteration, we compute $||f(\mathcal{X}_{old};\Theta_i) - f(\mathcal{X}_{old};\Theta_{i-1})||_2$, where $\mathcal{X}_{old}$ stands for memory images of old classes. Fig. 6 visualizes the feature distance of ER+CCL-DC and ER+ours throughout the entire training process. It is evident that our approach significantly minimizes the feature drift, resulting in a smoother curve.

6) Model Generalization Analysis: Based on [54], we adopt the flatness of the loss landscape for model generalization analysis. Fig. 7 visualizes the loss landscape of four students in CCL-DC and our method. For example, "Ours-S1" denotes the loss of the first student within our approach. In this figure, we employ the CE loss over all training samples of all learned tasks, while the axes represent the model parameters after PCA [55]. Initially, all models are situated in the regions with the lowest loss, denoted as red points. As more tasks arrive, all models remain within basins, but the final loss of CCL-DC is much larger than ours, indicating a better

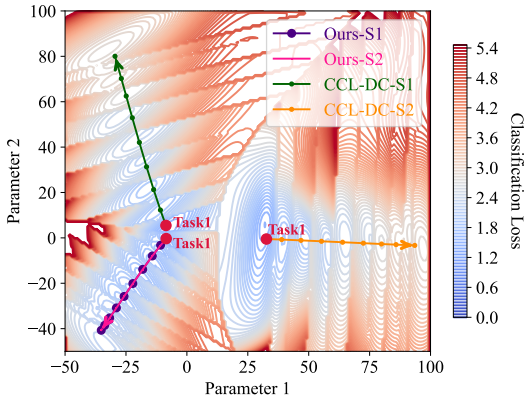


Fig. 7: Visualization of the loss landscape of students in CCL-DC and our method on CIFAR-100 at 1K memory. The axes depict the parameters after PCA while the dots symbolize the models after each task. The arrows indicate the direction of task processing. At Task 1, all models find their lowest loss at the red dots. As more tasks arrive, although all students reside in basins, the loss of our model stays almost constant.

generation of our model. Additionally, both students in our model consistently converge into the same basin, whereas CCL-DC exhibits significant divergence.

V. Conclusion

This study enhances ensemble learning by developing a Global Workspace Model (GWM) for OCIL. The overall framework consists of several student models and a GWM. By employing periodic parameter fusion, the GWM guides the students' learning process. Additionally, a Multi-level Collaborative Distillation strategy is devised to enforce peer-to-peer consistency between students and preserve historical knowledge. Extensive experiments on three popular OCIL benchmarks demonstrate the effectiveness of our method in enhancing stability while maintaining plasticity, resulting in notable improvements in overall performance. In the future, we plan to explore an increased number of heterogeneous collaborative learners to better simulate the competition-coordination dynamics suggested in cognitive systems. Additionally, we intend to enhance competitiveness by crafting effective data augmentation strategies that expand the range of student inputs, thereby boosting learning robustness and generalization.

References

[1] L. Wang, X. Zhang, H. Su, and J. Zhu, "A comprehensive survey of continual learning: theory, method and application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 8, pp. 5362–5383, 2024.

[2] Y. Ghunaim, A. Bibi, K. Alhamoud, M. Alfarra, H. A. Al Kader Hammoud, A. Prabhu, P. H. Torr, and B. Ghanem, "Real-time evaluation in online continual learning: A new hope," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 11 888–11 897.

[3] J. Gu, K. Wang, W. Jiang, and Y. You, "Summarizing stream data for memory-constrained online continual learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 11, 2024, pp. 12 217–12 225.

[4] J. Wu, S. Wang, Y. Sun, B. Yin, and Q. Huang, "Dual-domain division multiplexer for general continual learning: A pseudo causal intervention strategy," *IEEE Trans. Image Process.*, vol. 34, pp. 1966–1979, 2025.

[5] A. Soutif-Cormerais, A. Carta, A. Cossu, J. Hurtado, V. Lomonaco, J. Van de Weijer, and H. Hemati, "A comprehensive empirical evaluation on online continual learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3518–3528.

[6] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*. Elsevier, 1989, vol. 24, pp. 109–165.

[7] Y. Zhou, J. Yao, F. Hong, Y. Zhang, and Y. Wang, "Balanced destruction-reconstruction dynamics for memory-replay class incremental learning," *IEEE Trans. Image Process.*, vol. 33, pp. 4966–4981, 2024.

[8] D. Shim, Z. Mai, J. Jeong, S. Sanner, H. Kim, and J. Jang, "Online class-incremental continual learning with adversarial shapley value," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 11, 2021, pp. 9630–9638.

[9] A. Chaudhry, M. Rohrbach, M. Elhoseiny, T. Ajanthan, P. K. Dokania, P. H. Torr, and M. Ranzato, "On tiny episodic memories in continual learning," *arXiv preprint arXiv:1902.10486*, 2019.

[10] X. Jin, A. Sadhu, J. Du, and X. Ren, "Gradient-based editing of memory examples for online task-free continual learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 193–29 205, 2021.

[11] R. Aljundi, E. Belilovsky, T. Tuytelaars, L. Charlin, M. Caccia, M. Lin, and L. Page-Caccia, "Online continual learning with maximal interfered retrieval," *Advances in neural information processing systems*, vol. 32, 2019.

[12] Q. Wang, R. Wang, Y. Wu, X. Jia, and D. Meng, "Cba: Improving online continual learning via continual bias adaptor," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 19 082–19 092.

[13] H. Lin, B. Zhang, S. Feng, X. Li, and Y. Ye, "Pcr: Proxy-based contrastive replay for online class-incremental continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 24 246–24 255.

[14] Y. Guo, B. Liu, and D. Zhao, "Online continual learning through mutual information maximization," in *International conference on machine learning*. PMLR, 2022, pp. 8109–8126.

[15] L. Caccia, R. Aljundi, N. Asadi, T. Tuytelaars, J. Pineau, and E. Belilovsky, "New insights on reducing abrupt representation change in online continual learning," in *International Conference on Learning Representations*, 2022.

[16] G. Liang, Z. Chen, Z. Chen, S. Ji, and Y. Zhang, "New insights on relieving task-recency bias for online class incremental learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 5, pp. 3451–3464, 2024.

[17] Y. Wei, J. Ye, Z. Huang, J. Zhang, and H. Shan, "Online prototype learning for online continual learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18 764–18 774.

[18] M. Wang, N. Michel, L. Xiao, and T. Yamasaki, "Improving plasticity in online continual learning via collaborative learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 23 460–23 469.

[19] B. J. Baars, "Global workspace theory of consciousness: toward a cognitive neuroscience of human experience," *Progress in brain research*, vol. 150, pp. 45–53, 2005.

[20] S. Dehaene and J.-P. Changeux, "Experimental and theoretical approaches to conscious processing," *Neuron*, vol. 70, no. 2, pp. 200–227, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0896627311002583

[21] B. J. Baars, *In the Theater of Consciousness: The Workspace of the Mind*. Oxford University Press, 03 1997. [Online]. Available: https://doi.org/10.1093/acprof:oso/9780195102659.001.1

[22] J. Frankle, G. K. Dziugaite, D. Roy, and M. Carbin, "Linear mode connectivity and the lottery ticket hypothesis," in *Proceedings of the 37th International Conference on Machine Learning*, vol. 119. PMLR, 2020, pp. 3259–3269.

[23] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural computation*, vol. 9, no. 1, pp. 1–42, 1997.

[24] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware minimization for efficiently improving generalization," in *International Conference on Learning Representations*, 2021.

[25] Y. Gu, X. Yang, K. Wei, and C. Deng, "Not just selection, but exploration: Online class-incremental continual learning via dual view consistency," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7442–7451.

[26] Z. Wang, L. Liu, Y. Kong, J. Guo, and D. Tao, "Online continual learning with contrastive vision transformer," in *European Conference on Computer Vision*. Springer, 2022, pp. 631–650.

[27] S. Raghavan, J. He, and F. Zhu, "Online class-incremental learning for real-world food image classification," in *Proceedings of the IEEE/CVF*

Winter Conference on Applications of Computer Vision, 2024, pp. 8195–8204.

[28] H. Yan, L. Wang, K. Ma, and Y. Zhong, "Orchestrate latent expertise: Advancing online continual learning with multi-level supervision and reverse self-distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 23 670–23 680.

[29] Y. Guo, B. Liu, and D. Zhao, "Dealing with cross-task class discrimination in online continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 11 878–11 887.

[30] N. Michel, M. Wang, L. Xiao, and T. Yamasaki, "Rethinking momentum knowledge distillation in online continual learning," in *International Conference on Machine Learning*. PMLR, 2024, pp. 35 607–35 622.

[31] M. Seo, H. Koh, W. Jeung, M. Lee, S. Kim, H. Lee, S. Cho, S. Choi, H. Kim, and J. Choi, "Learning equi-angular representations for online continual learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 23 933–23 942.

[32] H. Lin, S. Feng, B. Zhang, H. Qiao, X. Li, and Y. Ye, "Uer: A heuristic bias addressing approach for online continual learning," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 96–104.

[33] N. Michel, G. Chierchia, R. Negrel, and J.-F. Bercher, "Learning representations on the unit sphere: Investigating angular gaussian and von mises-fisher distributions for online continual learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 13, 2024, pp. 14 350–14 358.

[34] Y. Wu, H. Wang, P. Zhao, Y. Zheng, Y. Wei, and L.-K. Huang, "Mitigating catastrophic forgetting in online continual learning by modeling previous task interrelations via pareto optimization," in *Forty-first International Conference on Machine Learning*, 2024, pp. 53 892–53 908.

[35] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *Advances in neural information processing systems*, vol. 33, pp. 18 661–18 673, 2020.

[36] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 3589–3599.

[37] S. Chen, M. Zhang, J. Zhang, and K. Huang, "Exemplar-based continual learning via contrastive learning," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 7, pp. 3313–3324, 2024.

[38] L. Wang and K.-J. Yoon, "Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3048–3068, 2021.

[39] S. Li, T. Su, X. Zhang, and Z. Wang, "Continual learning with knowledge distillation: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 6, pp. 9798–9818, 2025.

[40] K. Roy, C. Simon, P. Moghadam, and M. Harandi, "Subspace distillation for continual learning," *Neural Networks*, vol. 167, pp. 65–79, 2023.

[41] K. Li, J. Wan, and S. Yu, "Ckdf: Cascaded knowledge distillation framework for robust incremental learning," *IEEE Trans. Image Process.*, vol. 31, pp. 3825–3837, 2022.

[42] J. Lu and S. Sun, "Pamk: Prototype augmented multi-teacher knowledge transfer network for continual zero-shot learning," *IEEE Trans. Image Process.*, vol. 33, pp. 3353–3368, 2024.

[43] Z. Ji, J. Li, Q. Wang, and Z. Zhang, "Complementary calibration: Boosting general continual learning with collaborative distillation and self-supervision," *IEEE Trans. Image Process.*, vol. 32, pp. 657–667, 2022.

[44] H. Koh, M. Seo, J. Bang, H. Song, D. Hong, S. Park, J.-W. Ha, and J. Choi, "Online boundary-free continual learning by scheduled data prior," in *The Eleventh International Conference on Learning Representations*, 2023.

[45] Y.-n. Han and J.-w. Liu, "Online continual learning via the meta-learning update with multi-scale knowledge distillation and data augmentation," *Engineering applications of artificial intelligence*, vol. 113, p. 104966, 2022.

[46] B. Neyshabur, H. Sedghi, and C. Zhang, "What is being transferred in transfer learning?" *Advances in neural information processing systems*, vol. 33, pp. 512–523, 2020.

[47] S. P. Singh and M. Jaggi, "Model fusion via optimal transport," *Advances in Neural Information Processing Systems*, vol. 33, pp. 22 045–22 055, 2020.

[48] T. Zhang, M. Xue, J. Zhang, H. Zhang, Y. Wang, L. Cheng, J. Song, and M. Song, "Generalization matters: Loss minima flattening via parameter hybridization for efficient online knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 20 176–20 185.

[49] A. Krizhevsky, V. Nair, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., Apr. 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html

[50] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[51] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.

[52] G. Liang, Z. Chen, S. Su, S. Zhang, and Y. Zhang, "A masking, linkage and guidance framework for online class incremental learning," *Pattern Recognition*, vol. 160, p. 111185, 2025.

[53] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 2020, pp. 702–703.

[54] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=H1oyRlYgg

[55] A. Maćkiewicz and W. Ratajczak, "Principal components analysis (pca)," *Computers & Geosciences*, vol. 19, no. 3, pp. 303–342, 1993.