Memory-Aware Replay and Loss Balance for Long-Tailed Class Incremental Learning with Vision-Language Models

Xiasi Wang[2,*], Jiale Zheng[3], Jianfeng Zhang[3], Yi Huang[4], Lujia Pan[3], Runqi Wang[1,5,†]
[1] School of Computer Science and Technology, Beijing Jiaotong University, Beijing, China
[2]The Hong Kong University of Science and Technology
[3]Huawei Noah's Ark Lab [4]University of Chinese Academy of Sciences
[5]Beijing Key Laboratory of Traffic Data Mining and Embodied Intelligence, Beijing, China

Abstract

Class-incremental learning (CIL) requires a model to learn new classes sequentially without forgetting old ones. A key limitation of current VLM-based CIL methods lies in their assumption of balanced data—an assumption that fails in practice, leading to severe performance degradation and exacerbated forgetting when data follows a long-tailed distribution. In this work, we investigate the long-tailed class incremental learning (LT-CIL) problem within the vision-language framework of CLIP for the first time. Our proposed method, Memory-Aware Replay and Loss Balance, dubbed `MARBLE`, addresses the dual challenges of LT-CIL. To achieve a performance-efficiency trade-off, we introduce a memory-aware replay mechanism that strategically selects classes for rehearsal based on historical fragile class-wise performance and distribution statistics. Furthermore, to counteract the compounded data imbalance arising from both new task and replayed data, we employ an adaptive loss re-weighting strategy that dynamically balances the learning signal across head and tail classes. Extensive results show that our approach significantly outperforms existing benchmarks in diverse LT-CIL settings.

Keywords: Vision-Language Model, Class Incremental Learning, Long-Tailed Learning, Prompt Tuning

1. Introduction

Class-incremental learning (CIL) has emerged as a critical paradigm for enabling models with the ability of learning from continuously evolving environments [18, 36]. CIL requires a model to learn from a sequence of tasks where each task contains new classes, while retaining performance on all classes encountered in previous tasks. The core challenge in CIL is *catastrophic forgetting*, a phenomenon where the model's performance on old classes drastically deteriorates as it adapts to new data. Endeavors have been made to address this challenge, and existing methods can be broadly categorized into three groups: Regularization-based methods [18, 27, 39, 1, 36, 8, 29, 40], which impose constraints on parameter updates to protect acquired knowledge; Memory replay-based methods [2, 4, 24, 22, 5, 30], which retain a subset of past exemplars in a memory bank and replay them during new task training phases for rehearsing previously learned patterns; and Architecture expansion-based methods [23, 17, 20, 32, 10, 9, 38], which dynamically expand the network's capacity by adding new parameters or sub-networks for incoming tasks. Concurrently, the emergence of large-scale pre-trained vision-language models (VLMs), notably CLIP [26], has facilitated the extension of CIL into the multimodal domain. Pioneering studies such as [33, 14] have explored the use of prompt tuning [15] to adapt pretrained CLIP for incremental learning scenarios. These VLM-based CIL methods have demonstrated superior performance, which can be largely attributed to the highly transferable visual-semantic representations acquired during large-scale pre-training. These representations provide the model with a robust prior for visual recognition, establishing a strong foundation for CIL.

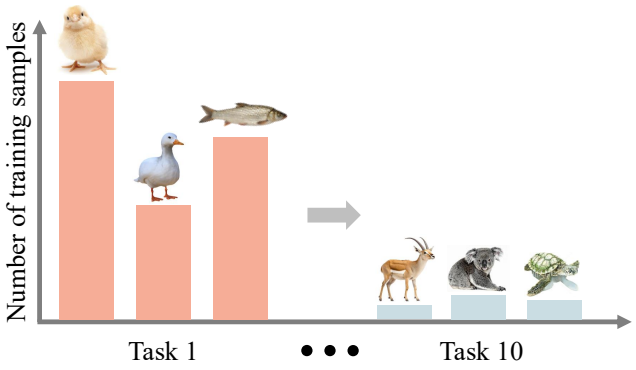

Figure 1. Illustration of long-tailed class incremental learning.

* Email to: xwangfy@connect.ust.hk
† Corresponding author

However, a common and often impractical assumption of existing works is that data within each incremental task is balanced across classes. This limits the practical applicability of these methods, as real-world data streams are always characterized by long-tailed distributions. For instance, a wildlife monitoring system learns to identify new species over time, but endangered species will have drastically fewer sightings than common species (illustrated in Figure 1). The more challenging and practical problem of long-tailed class-incremental learning (LT-CIL) remains largely unexplored within the VLM-based CLIP framework.

The LT-CIL scenario introduces a dual imbalance that exacerbates catastrophic forgetting. First, *inter-task imbalance* exists, where the volume of training data varies significantly across different tasks. Second, *intra-task* imbalance prevails, where the data within each task follows a long-tailed distribution (we present an illustration of two types of LT-CIL in Section 3.1). This results in a compounded effect: the model's tendency to favor head classes in the new task accelerates the forgetting of tail classes from previous tasks. Therefore, the long-tailed nature not only exists within tasks but also intensifies the inherent challenge of catastrophic forgetting across tasks. Recently, some works have studied the LT-CIL problem. For instance, [34] constructs sub-prototype space for knowledge integration and [12] re-weights gradients to pursue a balanced optimization. However, these approaches are confined to the traditional CIL framework, relying on convolutional neural networks trained from scratch, which overlooks the significant potential of large-scale pre-trained models like CLIP. Our work bridges this gap by studying how to leverage the robust prior knowledge of VLMs to develop a more effective and equitable learning system that is resilient to the dual challenges of long-tailed and incremental data.

To address the LT-CIL problem within the vision-language paradigm, we propose Memory-Aware Replay and Loss Balance, short as `MARBLE`. `MARBLE` has two major components to tackle catastrophic forgetting and data imbalance in LT-CIL concurrently. First, a memory bank is maintained to preserve historical exemplars for all seen historical classes. Furthermore, to navigate the performance-efficiency trade-off, we introduce a principled replay criterion that selects classes from the memory bank for rehearsal by jointly considering their historical performance (to identify fragile knowledge) and their distribution statistics (to harmonize the current task's imbalance). Second, to address the inter- and intra-task imbalance, we employ an adaptive loss re-weighting strategy to dynamically assign loss weights to samples of different classes, thereby balancing the gradient contributions from both head and tail classes. These two modules work in concert to provide a unified solution for the LT-CIL problem. The design of `MARBLE` is agnostic to specific CLIP finetuning methods, making it an orthogonal component that can be readily incorporated into established baselines to enhance their performance towards the LT-CIL challenge. Extensive experiments on CIFAR-100 and ImageNet-Subset demonstrate that our proposed method outperforms existing benchmarks by a large margin in the LT-CIL setting.

The contributions of our work are summarized as follows:

- We identify and investigate the challenging yet practical long-tailed class incremental learning problem within the vision-language model paradigm for the first time.
- We propose a novel framework `MARBLE`, which features memory-aware replay and adaptive loss balance modules to concurrently address the catastrophic forgetting and imbalance issues in the LT-CIL setting.
- We conduct extensive experiments on two benchmarks and achieved outperforming performances on multiple long-tailed settings compared with existing baselines.

2. Related Work

2.1. Long-Tailed Class Incremental Learning

Class incremental learning (CIL) is a subfield of continual learning. It aims to continuously learn knowledge from a stream of data with an expanding number of classes. The core challenge of CIL is *catastrophic forgetting*: the tendency for performance on old tasks to degrade when training on new ones [25].

Standard CIL. Existing class incremental learning methods can be categorized into three groups: regularization-based methods, memory replay-based methods, and architecture expansion-based methods. Regularization-based methods aim to constrain the change of the model's parameters. For example, [18, 27, 3, 36, 8, 29] apply knowledge distillation to the output logits or embeddings to maintain the consistency of model parameters. Memory replay-based methods adopt an external memory bank to store samples of historical tasks, and they are combined with the new task data for training to strengthen the model's performance on previous tasks. Some methods select samples based on class mean [2, 4] or via meta-learning[21]. Architecture expansion-based methods [23, 17, 20, 37, 32] gradually increase the size of the model to isolate the parameters for different tasks to ensure previously learned information is unaffected.

Long-Tailed CIL. In our work, we study the long-tailed CIL problem. Addressing both class incremental learning and class imbalance problems simultaneously is an intricate task since the imbalanced dataset exacerbates the catastrophic forgetting problem. Recently, some works have made endeavors towards this LT-CIL problem. [19]

first studies the LT-CIL problem and proposes a two-stage framework to address the imbalance problem. [34] constructs a sub-prototype space to integrate knowledge from different classes, and [12] investigates the gradients and proposes to reweight the gradients towards balanced optimization and unbiased classifier learning. However, these works are based on the traditional convolutional neural network backbone, where the model is trained from scratch. Our work differs from theirs in that we study the LT-CIL problem for pretrained vision-language models for the first time.

2.2. Prompt Tuning for Class Incremental Learning

Prompt tuning [15] is a popular method to fine-tune a pretrained model. It attaches learnable parameters as prompts during the training phase to adapt the pretrained model to downstream tasks. The advent of Vision-Language Models (VLMs) like CLIP [26] has demonstrated remarkable performance across various vision tasks. Some works have leveraged prompt tuning with CLIP for continual learning. For example, [33] firstly proposes the concept of textual attribute prompts to encode the common knowledge for different classes. [14] adopts a probabilistic modeling framework over visual-guided text features. However, all these works overlook the class-imbalanced data distribution problem, where the catastrophic forgetting problem in minority classes is exacerbated in this case. Our work simultaneously addresses the class-incremental learning and class imbalance within the CLIP framework.

3. Method

In this section, we first clarify the problem setting of the LT-CIL, and the preliminaries of CLIP prompt tuning. Then, we elaborate on the two key components in our proposed method MARBLE.

3.1. Preliminary

Long-Tailed Class Incremental Learning. The goal of class incremental learning is to learn from a sequentially arriving tasks $\{\mathcal{D}^1, \mathcal{D}^2, \cdots, \mathcal{D}^T\}$, without forgetting old knowledge. The intersected class label set between any two tasks is empty $\mathcal{C}^m \cap \mathcal{C}^n = \emptyset$, i.e., each task has its unique training classes. For the t-th task $\mathcal{D}^t = \{(x_i, y_i)\}_{i=1}^{n_t}$, the model trains on the n_t training classes with label set $\mathcal{C}^t$, while the data of previous tasks $\{\mathcal{D}^1, \mathcal{D}^2, \cdots, \mathcal{D}^{t-1}\}$ are no longer available. After the model is trained on task $\mathcal{D}^t$, it is evaluated on the test set of the current task and all previous tasks, i.e., the label set of the test set is $\mathcal{C}^1 \cup \cdots \cup \mathcal{C}^{t-1} \cup \mathcal{C}^t$. Moreover, our setting is task-agnostic, meaning the task ID is not accessible at inference time.

In this work, we specifically focus on the long-tailed CIL setting (LT-CIL), where the number of training data for different classes is imbalanced. Following previous works [19, 34], we consider two LT-CIL settings: Ordered LT-CIL and Shuffled LT-CIL. As illustrated in the Figure 2, they exhibit distinct data distributions. The ordered scenario constitutes a sequential structure where the sample size decreases monotonically across tasks, forming an explicit long tail. The shuffled scenario demonstrates a non-sequential pattern, as the training samples are randomly allocated, resulting in an irregular long-tail distribution.

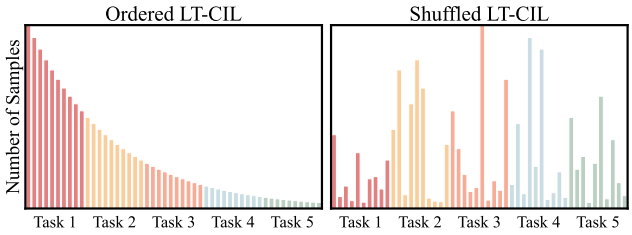

Figure 2. Illustration of Ordered and Shuffled LT-CIL.

Prompt Tuning for CLIP. CLIP [26] aligns visual and linguistic embeddings in a shared feature space. CLIP has an image encoder f_I and a text encoder f_T. For zero-shot classification, a set of hand-crafted prompts, typically like "a photo of a [CLASS]", are created by inserting each class name into the template. The text encoder converts these prompts into a set of vectors $\{tc_1, tc_2, \cdots, tc_K\}$ for K classes. The probability that an image x (encoded as $f_I(x)$) belongs to class k is given by:

$$p(y=k|x) = \frac{\exp(\text{sim}(f_I(x), tc_k)/\tau)}{\sum_{i=1}^{K} \exp(\text{sim}(f_I(x), tc_i)/\tau)}, \quad (1)$$

where the $\text{sim}(\cdot)$ is the cosine similarity and τ is a temperature parameter.

To further improve the performance of CLIP on downstream tasks, instead of using a fixed template, prompt tuning [41] introduces a set of continuous context vectors $\{p_1, p_2, \cdots, p_m\}$ as learnable parameters. Denote $tp_k = \text{concat}(p_1, p_2, \cdots, p_m, e_k)$, where e_k is the embedding of the class k's name. The class-specific text representation thus becomes $f_T(tp_k)$. These vectors are optimized on task-specific data to minimize the classification loss.

3.2. Overview of MARBLE

The illustration of MARBLE is shown in Figure 3. There are two key components in MARBLE. During the task sequences, we maintain a memory bank for storing historical exemplars. For the training of the current task, we balance the contribution of different classes by reweighting the loss per class (loss balance training). After training, we select samples from the memory bank (through memory-aware replay) for the rehearsal of the incoming task. For simplicity, we first elaborate on the memory-aware replay module for

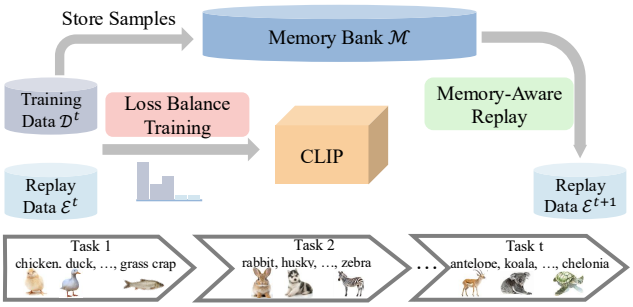

Figure 3. Illustration of MARBLE.

selecting data for rehearsal, and then introduce our loss balance training.

3.3. Memory-Aware Replay

Memory replay has been established as one of the most effective strategies to mitigate catastrophic forgetting in class-incremental learning. A common premise in conventional CIL settings is that the storage cost of maintaining a memory buffer is the primary constraint. Consequently, most existing methods operate under a fixed memory budget. However, within the vision-language framework of models like CLIP, we argue that storing exemplars for historical classes is relatively inexpensive, while the computational cost of replaying these samples during the training of new tasks becomes the dominant bottleneck. Motivated by this, we propose a memory-aware replay strategy. Instead of adhering to a fixed budget, we maintain a memory bank that preserves a few exemplars for all previously seen classes. Furthermore, we introduce a principled criterion specifically designed for LT-CIL to selectively choose a subset of classes from this memory bank for replay.

Formally, after learning the t-th task, we randomly sample S exemplars per class from the current task's class set $\mathcal{C}^t$ to form $\mathcal{H}^t$. We incorporate these sampled data into the memory bank $\mathcal{M}^{t+1} = \mathcal{M}^t \cup \mathcal{H}^t$. Thus, $\mathcal{M}^{t+1}$ contains exemplars from all seen classes $\mathcal{C}^{1:t} = \cup_{i=1}^{t} \mathcal{C}^i$. Concurrently, we evaluate the model on the memory bank $\mathcal{M}^{t+1}$ and record the class-wise accuracy for each class $k \in \mathcal{C}^{1:t}$, denoted as A_k^t. A lower accuracy A_k^t indicates more fragile knowledge of class k, suggesting a higher need for rehearsal in subsequent tasks.

Simultaneously, to quantify the long-tailed characteristic, we maintain a record of the number of training samples seen for each class k up to task t, denoted as n_k^t. The imbalance ratio for class k is defined as:

$$\gamma_k^t = \log(\frac{N^t}{n_k^t}), \tag{2}$$

where $N_t = \sum_{k \in \mathcal{C}^{1:t}} n_k^t$ is the total number of all training samples seen so far. A higher γ_k^t indicates that class k is more scarce in the training data and thus requires more attention.

To integrate these two factors, we design a selection indicator I_k^t for each class k by combining the fragility and the imbalance ratio:

$$I_k^t = (1 - A_k^t) \times \gamma_k^t. \tag{3}$$

This indicator is then normalized across all classes to form a probability distribution, dictating the probability of each class being selected in the memory bank $\mathcal{M}^{t+1}$ for replay in the next task:

$$P_k^t = \frac{I_k^t}{\sum_{j \in \mathcal{C}_{1:t}} I_j^t}. \tag{4}$$

To balance the effectiveness and computational cost during training, we set a maximum number of classes, R_{max}, to be replayed from $\mathcal{M}$ for each task. For example, for the new task $(t+1)$-th, we sample at most R_{max} classes from $\mathcal{C}^{1:t}$ according to the above probability distribution, and replay all the stored exemplars of these selected classes (denoted as $\mathcal{E}^{t+1}$) in the new task.

3.4. Loss Balance Training

During the training phase of the $(t+1)$-th new task, the training dataset is composed of replayed exemplars and the current task data, denoted as $\mathcal{E}^{t+1} \cup \mathcal{D}^{t+1}$. This combined dataset exhibits a compounded imbalance stemming from two sources: (1) the intrinsic long-tailed distribution present in the original new task data, as illustrated in Figure 2, and (2) the extrinsic imbalance introduced by the limited replayed data. This dual imbalance causes the gradient-based optimization to be biased, hindering the learning of diverse classes.

Inspired by prior work on addressing class imbalance [6], we adopt a re-weighting strategy to adjust the contribution of each class to the total loss. Formally, for the combined training set $\mathcal{E}^{t+1} \cup \mathcal{D}^{t+1}$, we denote n_k^{t+1} as the number of training data for class k. To alleviate the data imbalance issue, we assign a loss weight ω_k^{t+1} to class k based on the effective number of samples:

$$\omega_k^{t+1} = \frac{1-\beta}{1-\beta^{n_k^{t+1}}}, \tag{5}$$

where $\beta \in [0, 1)$ is a hyperparameter that controls the degree of overlap in the data representation. A class with a smaller effective number of samples ($\frac{1-\beta^{n_k^{t+1}}}{1-\beta}$) is assigned a higher weight ω_k^{t+1}.

Final Objective. Hence, the final objective for new task training can be formulated as follows:

$$\mathcal{L}_{final}^{t+1} = \sum_{(x_i, y_i) \in (\mathcal{E}^{t+1} \cup \mathcal{D}^{t+1})} \omega_{y_i} \mathcal{L}^{t+1}(x_i, y_i), \tag{6}$$

where $\mathcal{L}(x_i, y_i)$ is the per-sample loss:

$$\mathcal{L}^{t+1}(x_i, y_i) = -\log \frac{\exp(f_I(x_i), f_T(tp_{y_i}))/\tau}{\sum_{j=1}^{K} \exp(f_I(x_i), f_T(tp_j))/\tau}. \quad (7)$$

K is the total number of classes in $\mathcal{E}^{t+1} \cup \mathcal{D}^{t+1}$, and $f_T(tp_j) = f_T(\text{concat}(p_1, p_2, \cdots, e_j))$ is the text representation of class k, as elaborated in Section 3.1.

3.5. Complete Algorithm

The complete algorithm of our proposed MARBLE is summarized in Algorithm 1. For the t-th task, we train the model with the replay data $\mathcal{E}^t$ (obtained after the previous $(t-1)$-th task training) and t-th task training data $\mathcal{D}^t$. After loss balance training, we obtain the replay data $\mathcal{E}^{t+1}$ for the incoming $(t+1)$-th new task through the memory-aware replay module.

Algorithm 1 MARBLE

Input: Memory bank $\mathcal{M}^t$, Replay data for t-th task $\mathcal{E}^t$, t-th task training data $\mathcal{D}^t$

Output: New Memory Bank $\mathcal{M}^{t+1}$, Replay data for $(t+1)$-th task $\mathcal{E}^{t+1}$

Train CLIP with Eq. 6 ▷ Loss balance training
Randomly sample S exemplars per class from $\mathcal{D}^t$ as $\mathcal{H}^t$. Obtain the new memory bank $\mathcal{M}^{t+1} = \mathcal{M}^t \cup \mathcal{H}^t$
for each class k in $\mathcal{M}^{t+1}$ **do** ▷ Memory-aware replay
    Calculate indicator $\mathcal{I}_k^t$ by Eq. 3
end for
Sample R_{max} classes from class label set $\mathcal{C}^{1:t}$ by Eq. 4
return Replay dataset $\mathcal{E}^{t+1}$

4. Experiments

In this section, we first elaborate on our experiment settings. Then, we compare our proposed MARBLE with baselines in diverse LT-CIL settings. Finally, we conduct ablation studies of key components of MARBLE to evaluate their effectiveness.

4.1. Experiments Settings

Datasets. Following previous work [33], we perform our experiments on two standard benchmarks, including CIFAR-100 [16] and ImageNet-Subset with 100 classes [7]. Each benchmark contains 100 classes in total. In our main results, we split them into 5/10 tasks, where each task contains 20/10 classes. We create the long-tail CIL setting by deleting samples for some classes according to different imbalance rates $\rho \in \{0.1, 0.01, 0.001\}$. The imbalance rate can be formulated as $\rho = \frac{n_{\min}}{n_{\max}}$, where $n_{\max}$ and $n_{\min}$ are the maximum and minimum number of training data for all classes respectively.

Evaluation Metrics. We use the standard metric in class incremental learning to measure the performance: average accuracy. More specifically, after the model is trained on the i-th task, we evaluate the model's performance on the test set of all previous tasks. Formally, we denote the i-th task trained model's accuracy on the j-th task ($j \leq i$) as $a_{i,j}$. In our main results, we split the whole benchmark into $T = 5/10$ tasks. Hence, the reported metric is:

$$\bar{a}_T = \sum_{j=1}^{T} a_{T,j}.$$

Compared Baselines. Our method is orthogonal to the CLIP-based finetuning methods, including prompt tuning. Hence, to evaluate the effectiveness and flexibility of our method, we combine MARBLE with existing methods, including CoOp [41] and AttriCLIP [33]. CoOp adopts the same prompts for all the training data, and AttriCLIP adopts an attribute bank and proposes to adaptively choose different prompts for each training data. We additionally provide the zero-shot results of Continual-CLIP [31] for a more comprehensive comparison.

Implementation Details. For Continual-CLIP, CoOp and AttriCLIP, we adopt the ViT-L-14 [26] as the backbone. All methods are evaluated under the task-agnostic setting, meaning the task ID is not known during testing. For each task, we train the model for 5 epochs, and adopt the SGD and optimizer with the learning rate to be 0.001. For the memory-aware replay, we store $S = 20$ exemplars per class into the memory bank. For replay, we select at most $R_{max} = 40$ classes of samples. In loss balance training, we set the β as 0.9 for all benchmarks.

4.2. Main Results

We evaluate our method in diverse long-tailed CIL settings. The results are reported in Table 1 and Figure 4.

Intensified Forgetting in LT-CIL. Our results first reveal a pronounced phenomenon of intensified catastrophic forgetting under the LT-CIL setting. As shown in Table 1, standard fine-tuning baselines (CoOp, AttriCLIP) yield only marginal gains or even underperform the zero-shot Continual-CLIP baseline. In Figure 4, we observe a severe accuracy drop of these baselines. This clearly indicates that their effectiveness is significantly hampered by the long-tailed data distribution, underscoring the unique challenge posed by LT-CIL.

Results of Ordered LT-CIL. As presented in Table 1 (upper part), our method consistently enhances the performance of existing baselines across various imbalance ratios. The improvement is particularly pronounced in certain cases. For instance, when integrated with AttriCLIP on ImageNet-Subset ($\rho = 0.1$, $T = 10$ tasks), our approach elevates the average accuracy to 82.5% from the

Methods	CIFAR-100						ImageNet-Subset					
	$\rho = 0.1$		$\rho = 0.01$		$\rho = 0.001$		$\rho = 0.1$		$\rho = 0.01$		$\rho = 0.001$	
	5 tasks	10 tasks	5 tasks	10 tasks	5 tasks	10 tasks	5 tasks	10 tasks	5 tasks	10 tasks	5 tasks	10 tasks
Ordered LT-CIL												
Continual-CLIP	66.7	66.7	66.7	66.7	66.7	66.7	70.1	70.1	70.1	70.1	70.1	70.1
CoOp	70.6	70.5	75.4	75.4	72.9	75.8	80.9	75.3	76.1	80.2	71.5	79.3
CoOp+MARBLE	80.2 +9.6	80.5 +10.0	79.9 +4.5	79.7 +4.3	76.7 +3.8	77.9 +2.1	83.0 +2.1	83.9 +8.5	85.0 +8.9	83.7 +3.5	81.4 +9.9	80.8 +1.5
AttriCLIP	71.8	72.7	73.3	73.4	73.0	73.6	80.1	70.1	76.5	72.7	72.3	76.1
AttriCLIP+MARBLE	80.5 +8.7	80.2 +7.5	80.4 +7.1	79.7 +6.3	77.3 +4.3	78.0 +4.4	83.6 +3.5	82.7 +12.6	84.5 +8.0	84.5 +11.8	78.9 +6.6	78.2 +2.1
Shuffled LT-CIL												
Continual-CLIP	66.7	66.7	66.7	66.7	66.7	66.7	70.1	70.1	70.1	70.1	70.1	70.1
CoOp	68.2	69.3	69.8	73.5	70.3	70.8	72.9	73.2	71.4	71.7	70.1	70.2
CoOp+MARBLE	78.9 +10.7	79.8 +10.5	78.7 +8.9	79.8 +6.3	73.6 +3.3	75.8 +5.0	81.3 +8.4	82.0 +8.8	80.6 +9.2	81.5 +9.8	79.2 +9.1	80.6 +10.4
AttriCLIP	69.1	70.7	69.2	70.8	66.2	69.3	76.2	75.3	71.0	74.2	66.5	72.6
AttriCLIP+MARBLE	79.0 +9.9	80.1 +9.4	78.5 +8.7	79.8 +9.0	75.8 +9.6	78.1 +8.8	86.0 +9.8	81.8 +6.5	80.6 +9.6	82.6 +8.4	76.6 +10.1	79.2 +6.6

Table 1. Average accuracy (%) for Ordered and Shuffled LT-CIL settings. "$T = 5/10$ tasks" means that we split the dataset into T tasks, and we report the average accuracy of all tasks after the model is trained on the last task. The red numbers are the absolute increase over their baselines.

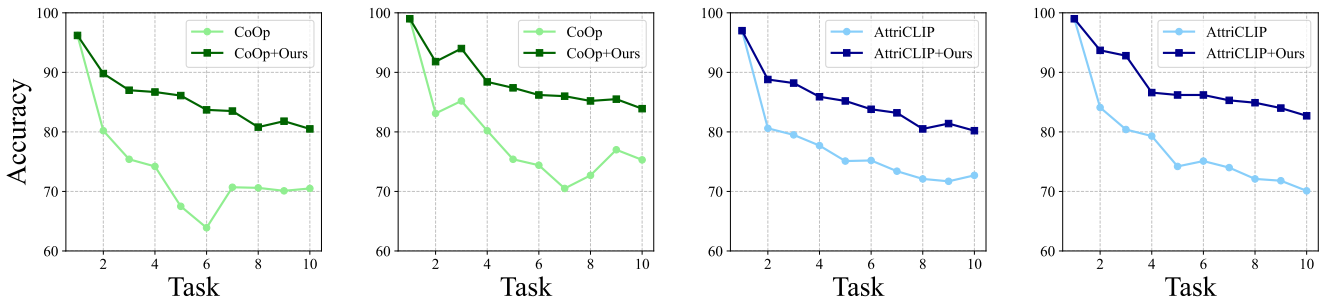

Figure 4. The accuracies (%) after each task (task 1 to task 10). The first and second graphs show the results for CIFAR-100 and ImageNet-Subset for CoOp-based methods, respectively, and the third and fourth graphs show the CIFAR-100 and ImageNet-Subset results for AttriCLIP-based methods, respectively. For all results, the evaluated setting is Ordered LT-CIL ($\rho = 0.1$).

baseline 70.1%, representing a substantial gain. Figure 4 provides the accuracies over the complete process of sequential tasks of Ordered LT-CIL. We can find that for baselines, the accuracy drops significantly, especially for the first two tasks, while the drop of our method is more stable and slower, demonstrating our method's effectiveness for anti-forgetting in the LT-CIL setting.

Results of Shuffled LT-CIL. We present the results of Shuffled LT-CIL settings in Table 1 (lower part). We can observe that our method brings consistent improvements when combined with baselines. Moreover, we find that for baselines, the results of the Shuffled LT-CIL setting are worse than those of the Ordered LT-CIL. This may due to the fact that the intra-task imbalance is more severe in Shuffled LT-CIL than Ordered LT-CIL. Our method narrows this performance gap by reweighting the losses of different classes.

4.3. Ablation and More Studies

We discuss the effectiveness of each part in our proposed method in this part. All the results in this section are conducted on CIFAR-100 with $T = 10$ tasks if not specifically stated. We present the results of CoOp+MARBLE for Ordered LT-CIL with different imbalance ratios $\rho \in \{0.1, 0.01, 0.001\}$.

Ablation Study of Two Modules. MARBLE comprises two major modules: memory-aware replay (*MAR*) and loss balance training (*LBT*). Here, we study the individual impact of these two modules. The results are shown in Table 2.

We can observe that with *MAR* module alone, our method achieves competitive results compared with the baseline. To further examine the effectiveness of the *LBT* module, we integrate *MAR* with existing methods designed to address the long-tail classification problem, including reweighting with inverse class frequency [13, 35], and resampling [28, 11]. It shows that when *MAR* is combined with inverse reweight, it achieves better performance than *MAR* alone, while resampling brings negligible improvement. The collaboration of *MAR* and *LBT* maximizes the effectiveness of our proposed MARBLE.

Methods	0.1	0.01	0.001	Avg.
CoOp	70.5	75.4	75.8	73.9
MAR	76.1	77.4	76.5	76.7
MAR+Inverse Reweight	79.0	78.6	77.2	78.3
MAR+Resample	77.3	76.8	76.4	76.8
MAR+*LBT* (Ours)	80.5	79.7	77.9	**79.4**

Table 2. Ablation results (%) with different imbalance ratios ρ.

Replay Data Selection Criteria. In our Memory-Aware Replay module, class selection is governed by a composite indicator (Eq. 3) that integrates class fragility and imbalance ratio. To evaluate the contribution of each criterion, we conduct an ablation study. Results in Table 3 demonstrate that while each individual criterion outperforms random selection, their combination yields synergistic effects and leads to superior performance.

Criteria	0.1	0.01	0.001	Avg.
Random	78.7	76.2	75.3	76.7
Fragility	79.8	79.1	77.1	78.7
Imbalance	79.5	79.2	76.9	78.5
Combined	80.5	79.7	77.9	**79.4**

Table 3. Results (%) for different class selection criteria.

Number of Sampled Classes and Stored Samples. In Figure 5 (left) and Table 4, we study the impact of the number of replayed classes R_{max}, reporting both accuracy and efficiency. The results indicate that as R_{max} increases, the GPU memory usage and training time increases, and when R_{max} is beyond 40, the accuracy stops increasing, suggesting a saturation point in performance. We therefore set $R_{max} = 40$ to strike a balance between performance and efficiency. Similarly, setting $S = 20$ achieves optimal performance, while overly small values (e.g., $S = 1$) lead to limited improvement due to insufficient representation of replayed classes. Based on these observations, we set $R_{max} = 40$ and $S = 20$ in our experiments.

Reweighting Parameter. In our loss-balancing module, we employ a re-weighting strategy parameterized by β (Eq. 5). As summarized in Table 5, the optimal performance is achieved when $\beta = 0.9$. As $\beta \to 1$, the weighting converges to inverse class frequency balancing [13, 35], while as $\beta \to 0$, it reduces to uniform weighting (i.e., no re-weighting). The superior result at $\beta = 0.9$ suggests a balanced compromise between these two strategies is most effective in our setting.

R_{max}	10	20	40	50
Accuracy (%)	77.5	77.3	79.4	79.3
GPU memory (MiB)	10686	13438	18850	21138
Training time (min)	73.8	74.4	78.9	82.8

Table 4. Accuracy and efficiency results for different number of replayed classes R_{max}.

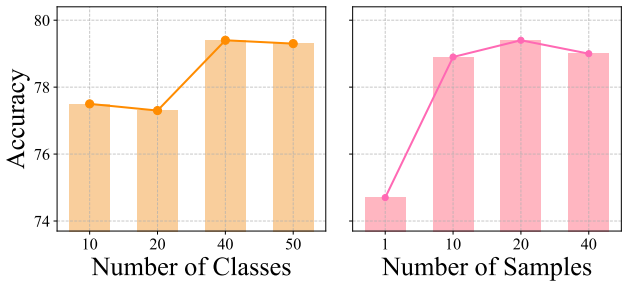

Figure 5. Results (%) for different number of replayed classes (left) and stored samples per class (right). We report the average result of three imbalance ratios.

β	0.1	0.01	0.001	Avg.
0.8	78.5	78.7	77.1	78.1
0.9	80.5	79.7	77.9	**79.4**
0.95	79.7	79.1	77.9	78.9
0.99	79.3	79.8	76.7	78.6

Table 5. Results (%) for different β.

Model Similarity Analysis. To better visualize the effect of MARBLE, we compute the Euclidean distance between the learned prompts after training on all sequential tasks. As illustrated in Figure 6, the prompt parameters of our method exhibit closer proximity compared to the baseline CoOp. This observation provides evidence that our approach enhances knowledge retention from previous tasks, resulting in more stable prompt learning throughout the incremental learning process.

Discussion on Generalizability. MARBLE comprises two independent modules: Memory-Aware Replay (*MAR*), which performs data-level sample selection for replay, and Loss-Balance Training (*LBT*), which performs loss-level reweighting to mitigate class imbalance. In contrast,

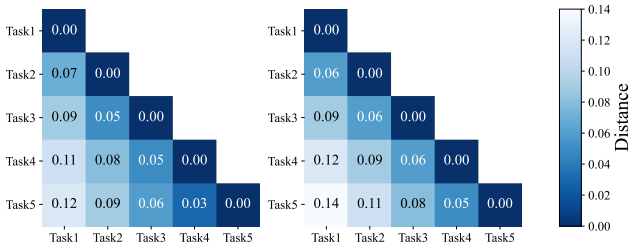

Figure 6. Prompt distance among models. Left figure is for CoOp+MARBLE, and right figure is for CoOp. The setting is Ordered LT with $T = 5$ tasks and $\rho = 0.1$.

fine-tuning methods (e.g., prompt-based or adapter-based) focus on model-level adaptation. *MAR* and *LBT* do not alter the model architecture; rather, *LBT* adjusts the classification loss and can be seamlessly integrated into the training objective of existing fine-tuning approaches. Therefore, MARBLE is orthogonal to these methods and holds the potential to improve performance in long-tailed class-incremental learning when properly combined.

5. Conclusion

In this work, we tackle the challenging yet underexplored long-tailed class-incremental learning problem within the vision-language paradigm for the first time. We propose MARBLE, a novel framework that concurrently addresses catastrophic forgetting and data imbalance through two key components: a memory-aware replay strategy for selective historical knowledge replay, and an adaptive loss-balancing training mechanism. The collaborative operation of these modules effectively mitigates the adverse effects of imbalance and forgetting, leading to significant performance gains in diverse LT-CIL settings. This work highlights the indispensability of addressing data imbalance for CIL, taking a step towards more robust and reliable continual learning systems.

Limitations. We acknowledge several limitations of this work. First, our evaluation primarily validates the performance of MARBLE integrated with a limited number of adaptation methods on CLIP, leaving it to be experimentally validated for its effectiveness when combined with a broader range of VLMs and their diverse fine-tuning approaches. Second, although justified by performance gains, the computational overhead introduced by our selective replay mechanism may be restrictive in more rigorous resource-constrained settings. Addressing these limitations offers promising directions for future research.

Acknowledgement

This work was partly supported by the National Key Research and Development Program of China under Grant 2024YFE0202900; the National Natural Science Foundation of China (62506028); Postdoctoral Innovation Talent Support Program (K25M200080); and Beijing Jiaotong University "Jingying Plan" No. 2024XKRC090.

Disclosure of Interests

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154, 2018. 1

[2] P. Buzzega, M. Boschini, A. Porrello, D. Abati, and S. Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020. 1, 2

[3] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 233–248, 2018. 2

[4] H. Cha, J. Lee, and J. Shin. Co2l: Contrastive continual learning. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 9516–9525, 2021. 1, 2

[5] S. Channappayya, B. R. Tamma, et al. Augmented memory replay-based continual learning approaches for network intrusion detection. *Advances in Neural Information Processing Systems*, 36:17156–17169, 2023. 1

[6] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019. 4

[7] 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*, pages 248–255. Ieee, 2009. 5

[8] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision*, pages 86–102. Springer, 2020. 1, 2

[9] A. Douillard, A. Ramé, G. Couairon, and M. Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9285–9295, 2022. 1

[10] Q. Gao, Z. Luo, D. Klabjan, and F. Zhang. Efficient architecture search for continual learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11):8555–8565, 2022. 1

[11] Y. Geifman and R. El-Yaniv. Deep active learning over the long tail. *arXiv preprint arXiv:1711.00941*, 2017. 7

[12] J. He. Gradient reweighting: Towards imbalanced class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16668–16677, 2024. 2, 3

[13] C. Huang, Y. Li, C. C. Loy, and X. Tang. Learning deep representation for imbalanced classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5375–5384, 2016. 7

[14] S. Jha, D. Gong, and L. Yao. Clap4clip: Continual learning with probabilistic finetuning for vision-language models. *Advances in neural information processing systems*, 37:129146–129186, 2024. 1, 3

[15] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim. Visual prompt tuning. In *European conference on computer vision*, pages 709–727. Springer, 2022. 1, 3

[16] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[17] X. Li, Y. Zhou, T. Wu, R. Socher, and C. Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International conference on machine learning*, pages 3925–3934. PMLR, 2019. 1, 2

[18] Z. Li and D. Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 1, 2

[19] X. Liu, Y.-S. Hu, X.-S. Cao, A. D. Bagdanov, K. Li, and M.-M. Cheng. Long-tailed class incremental learning. In *European Conference on Computer Vision*, pages 495–512. Springer, 2022. 2, 3

[20] Y. Liu, B. Schiele, and Q. Sun. Adaptive aggregation networks for class-incremental learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 2544–2553, 2021. 1, 2

[21] Y. Liu, Y. Su, A.-A. Liu, B. Schiele, and Q. Sun. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 12245–12254, 2020. 2

[22] 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*, pages 3589–3599, 2021. 1

[23] A. Mallya and S. Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7765–7773, 2018. 1, 2

[24] A. Maracani, U. Michieli, M. Toldo, and P. Zanuttigh. Recall: Replay-based continual learning in semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7026–7035, 2021. 1

[25] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019. 2

[26] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 1, 3, 5

[27] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 1, 2

[28] L. Shen, Z. Lin, and Q. Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *European conference on computer vision*, pages 467–482. Springer, 2016. 7

[29] C. Simon, P. Koniusz, and M. Harandi. On learning the geodesic path for incremental learning. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1591–1600, 2021. 1, 2

[30] J. S. Smith, L. Valkov, S. Halbe, V. Gutta, R. Feris, Z. Kira, and L. Karlinsky. Adaptive memory replay for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3605–3615, 2024. 1

[31] V. Thengane, S. Khan, M. Hayat, and F. Khan. Clip model is an efficient continual learner. *arXiv preprint arXiv:2210.03114*, 2022. 5

[32] F.-Y. Wang, D.-W. Zhou, H.-J. Ye, and D.-C. Zhan. Foster: Feature boosting and compression for class-incremental learning. In *European conference on computer vision*, pages 398–414. Springer, 2022. 1, 2

[33] R. Wang, X. Duan, G. Kang, J. Liu, S. Lin, S. Xu, J. Lü, and B. Zhang. Attriclip: A non-incremental learner for incremental knowledge learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3654–3663, 2023. 1, 3, 5

[34] X. Wang, X. Yang, J. Yin, K. Wei, and C. Deng. Long-tail class incremental learning via independent sub-prototype construction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28598–28607, 2024. 2, 3

[35] Y.-X. Wang, D. Ramanan, and M. Hebert. Learning to model the tail. *Advances in neural information processing systems*, 30, 2017. 7

[36] Y. Wu, Y. Chen, L. Wang, Y. Ye, Z. Liu, Y. Guo, and Y. Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 374–382, 2019. 1, 2

[37] S. Yan, J. Xie, and X. He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3014–3023, 2021. 2

[38] F. Ye and A. G. Bors. Self-evolved dynamic expansion model for task-free continual learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22102–22112, 2023. 1

[39] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pages 3987–3995. PMLR, 2017. 1

[40] X. Zhao, H. Wang, W. Huang, and W. Lin. A statistical theory of regularization-based continual learning. *arXiv preprint arXiv:2406.06213*, 2024. 1

[41] K. Zhou, J. Yang, C. C. Loy, and Z. Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022. 3, 5