PRELIMINARY PREPRINT VERSION – DO NOT CITE
This is not the version of record. The final version will appear in the Proceedings of IJCAI-ECAI 2026.

MOBO: A Merging-Oriented Bi-Level Optimization Framework for Class Incremental Learning

Siyu Zhang[1] , **Wen Wang**[1] , **Wenju Sun**[1] , **Qingyong Li**[1,2] and **Yangliao Geng**[1*]
[1]Key Laboratory of Big Data & Artificial Intelligence in Transportation (Ministry of Education), Beijing Jiaotong University
[2]Frontiers Science Center for Smart High-speed Railway System, Beijing Jiaotong University
{zhangsiyu, wangwen, sunwenju, liqy}@bjtu.edu.cn, gyarenas@gmail.com

Abstract

Class-Incremental Learning (CIL) aims to enable models to sequentially learn new tasks while retaining knowledge from previous ones. Recently, merging-based pre-trained CIL methods have gained significant attention due to their competitive performance and high inference efficiency. However, most existing approaches decouple training from merging, neglecting the compatibility among task-specific adapters. This incompatibility introduces severe conflicts during integration, resulting in catastrophic forgetting and notable performance degradation. To address this limitation, we propose MOBO, a Merging-Oriented Bi-Level Optimization framework that synergizes the optimization of the current task model with the performance of the final merged model. At the upper level, we introduce a global loss that anticipates the merged model's behavior, guiding the current task parameters toward a solution space that facilitates effective merging. At the lower level, we employ a dynamic weighted merging strategy to optimize merging coefficients and update the merged model. By alternately optimizing the task-specific and merged models, MOBO effectively mitigates the performance loss caused by incompatible adapter integration. Comprehensive experiments on CIFAR-100, CUB-200, ImageNet-R, ImageNet-A, and VTAB demonstrate the superiority of our approach, highlighting its robustness and scalability, particularly on long task sequences. Code is available at https://github.com/ZhangSY9979/MOBO.git

1 Introduction

In many open-world applications, data arrive sequentially and the set of classes expands over time, requiring learning systems to incrementally acquire new class knowledge without revisiting previous data [Thrun, 1995]. This paradigm, known as Class-Incremental Learning (CIL) [Rebuffi *et al.*, 2017], faces the fundamental challenge of catastrophic forgetting, where adaptation to new tasks disrupts representations learned for earlier classes. Recent advances in Pre-Trained Models (PTMs) [Wang *et al.*, 2022c] have inspired PTM-based CIL methods, which freeze the pretrained backbone and optimize lightweight components—such as prompts or adapters—to adapt to incremental tasks. By leveraging the generalizable representations acquired from large-scale data, PTM-based CIL approaches have demonstrated superior performance compared to models trained from scratch [Zhou *et al.*, 2024b; Zhou *et al.*, 2024c]. However, maintaining an ever-growing set of task-specific adapters introduces significant scalability challenges in terms of memory and deployment, especially over long task sequences [Sun *et al.*, 2025].

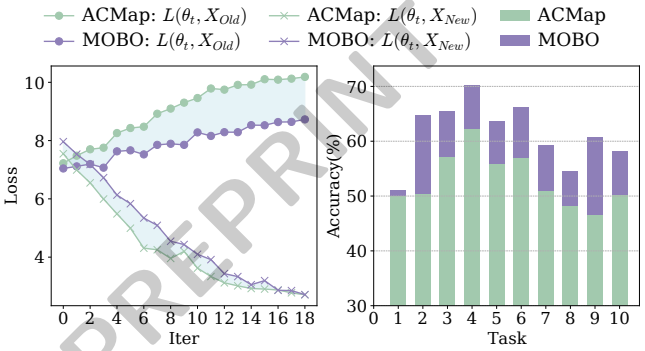

Figure 1: Comparison between the decoupled training method (ACMap) and our proposed bi-level optimization framework (MOBO) on the ImageNet-A dataset. **Left:** The evolution of loss for both new and old tasks during the current training phase. **Right:** The final performance of each task after training.

To mitigate this overhead, merging-based pre-trained CIL has emerged as a promising direction. Instead of maintaining separate task-specific modules at inference time, these methods consolidate them into a single model by merging parameters or adapter weights, achieving competitive accuracy with high inference efficiency. For instance, ACMap [Fukuda *et al.*, 2025] utilizes a simple average-based merging strategy to curb parameter growth while preserving performance, while PM [Qiu *et al.*, 2025] investigates the computation of optimal merging coefficients via a closed-form solution. Despite their effectiveness, most merging-based methods follow a decou-

*Corresponding author.

IJCAI-ECAI 2026 PREPRINT

PRELIMINARY PREPRINT VERSION – DO NOT CITE
This is not the version of record. The final version will appear in the Proceedings of IJCAI-ECAI 2026.

pled pipeline: they first train a task-specific adapter optimized primarily for the current task, then post hoc seek optimal coefficients to merge it into the previously learned model.

A key limitation of this decoupling is that successful merging depends not only on the merging coefficients but also on the *compatibility* among the models being merged. When task-specific adapters are trained independently, their parameters can drift into divergent regions of the solution space. Such incompatibility can induce severe conflicts during integration, leading to substantial performance degradation and exacerbated forgetting. As illustrated in Figure 1, this paradigm (represented by ACMap) may cause the new task model's loss on previous classes to increase rapidly, ultimately harming the final merged model. These observations motivate a natural question: *Can we explicitly optimize the new task parameters to be "merge-friendly" during the training phase?*

To address this issue, we propose a merging-oriented bi-level optimization framework for class-incremental learning (MOBO). The core idea is to synergize the learning of new tasks with the preservation of the global merged model. We formulate the problem as a bi-level optimization process. At the upper level, we introduce a merging-oriented global loss that anticipates the merged model's performance and guides the optimization of current task parameters toward merge-friendly solutions. At the lower level, we adopt a dynamic weighted merging strategy that optimizes merging coefficients and updates the merged model. By alternating between these two stages, MOBO jointly encourages effective learning on the new task and compatibility with the previously learned model, thus mitigating catastrophic forgetting caused by conflicting integrations.

The main contributions of this paper are summarized as follows:

- We propose MOBO, a bi-level optimization framework for CIL that synergizes the training of task-specific adapters with the integration of the merged model, effectively resolving the incompatibility issues inherent in decoupled approaches.
- We introduce a merging-oriented global loss tailored for the exemplar-free setting. By decomposing the objective into manageable sub-losses, MOBO guides the optimization trajectory toward a "merge-friendly" region without requiring access to historical data.
- Extensive experiments on CIFAR-100, CUB-200, ImageNet-R, ImageNet-A, and VTAB demonstrate that MOBO outperforms state-of-the-art methods and exhibits superior robustness and scalability, particularly in scenarios with long task sequences.

2 Related Work

Class-incremental learning (CIL) aims to enable models to continuously learn from new task data while preserving performance on previously learned tasks [Ratcliff, 1990]. The primary challenge in CIL is catastrophic forgetting, where the adaptation to new tasks degrades previously acquired knowledge. Existing approaches typically fall into four representative paradigms. Replay-based methods (e.g., [Aljundi *et al.*, 2019b; Chaudhry *et al.*, 2019]) mitigate forgetting by storing and replaying a subset of samples from previous tasks, thereby recovering historical knowledge during training [Liu *et al.*, 2020]. Knowledge distillation methods (e.g., [Dhar *et al.*, 2019; Douillard *et al.*, 2020]) transfer knowledge by aligning the representations of the updated model with those of the old model [Li and Hoiem, 2017; Rebuffi *et al.*, 2017]. Parameter regularization methods (e.g., [Aljundi *et al.*, 2018; Aljundi *et al.*, 2019a]) penalize changes to parameters critical for prior tasks, thus limiting parameter drift [Kirkpatrick *et al.*, 2017]. Finally, model correction methods (e.g., [Belouadah and Popescu, 2019; Pham *et al.*, 2022]) aim to rectify the prediction bias towards new classes to ensure balanced learning [Pian *et al.*, 2023]. Despite their effectiveness, these approaches often rely on assumptions that are difficult to satisfy in real-world scenarios, such as access to raw historical data or significant storage and computational resources.

Recently, CIL based on pre-trained models (PTMs) has emerged as a prominent research direction. In this setting, parameter-efficient fine-tuning (PEFT) methods [Han *et al.*, 2024] are widely adopted to mitigate catastrophic forgetting while preserving the robust representational capacity of PTMs. Prompt-based methods guide the PTM's behavior via learnable prompts; for instance, L2P [Wang *et al.*, 2022c] utilizes a prompt pool with key–query matching, while DualPrompt [Wang *et al.*, 2022b] incorporates global and expert prompts to encode task-shared and task-specific knowledge. Adapter-based methods introduce lightweight modules for adaptation, such as EASE [Zhou *et al.*, 2024c], which employs dynamic subspace expansion and prototype synthesis. MOS [Sun *et al.*, 2025] further optimize the training strategy for task-specific adapters by leveraging prior knowledge from the pretrained model to enhance performance. To further optimize inference efficiency, merging-based strategies have recently been proposed to consolidate task-specific modules into a single model. ACMap [Fukuda *et al.*, 2025] utilizes a simple average-based merging strategy to curb parameter growth while preserving performance. PM [Qiu *et al.*, 2025] extends this direction by investigating the computation of optimal merging coefficients via a closed-form solution. However, these methods typically follow a decoupled pipeline—training task-specific adapters first and merging them post-hoc—which often overlooks the parameter compatibility required for effective integration.

3 Preliminaries

3.1 Problem Setting

Consider a sequence of classification tasks with disjoint datasets $\{D_t\}_{t=1}^{T}$, where each $D_t = \{(X_t, \mathcal{Y}_t)\}$ consists of input–label pairs. The label spaces are non-overlapping across tasks, i.e., $\mathcal{Y}_t \cap \mathcal{Y}_{t'} = \emptyset$ for all $t \neq t'$. Following the standard CIL protocol, once training on task t is completed, data from previous tasks are no longer accessible. Moreover, at inference time, the task identity of a query is unknown (task-ID free); the model is required to classify samples over the union of all classes $\bigcup_{t=1}^{T} \mathcal{Y}_t$.

IJCAI-ECAI 2026 PREPRINT

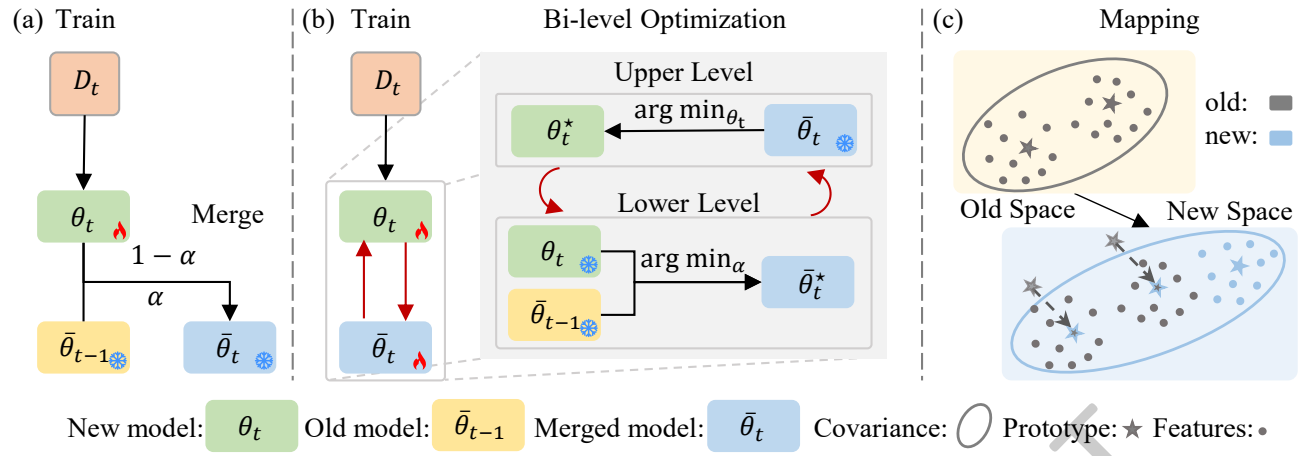

Figure 2: (a) The training procedure of ACMap. (b) The training procedure of our proposed MOBO, which adopts a bi-level optimization framework to alternately optimize the new task parameters θ_t and the merged model parameters $\bar{\theta}_t$. (c) Classifier optimization, which incrementally updates class prototypes and covariance matrices to achieve consistent modeling of class distributions across tasks.

3.2 Pre-Trained Model Based CIL

Recent state-of-the-art CIL approaches are predominantly built upon pre-trained models. They typically leverage a frozen pre-trained backbone as the encoder, along with trainable, randomly initialized linear classifiers [Wang *et al.*, 2022c]. Formally, let $F(\cdot;\Phi) : \mathbb{R}^D \to \mathbb{R}^d$ denote the pre-trained encoder with frozen parameters Φ, and let $W \in \mathbb{R}^{d\times|\mathcal{Y}|}$ denote the trainable classifier weights. For an input query x, the prediction is given by

$$f(x) = W^\top F(x,\theta). \tag{1}$$

Beyond training only the classifier, some methods further leverage parameter-efficient fine-tuning modules, such as adapters and prompts. A standard adapter follows a bottleneck architecture consisting of a down-projection $W_{\text{down}} \in \mathbb{R}^{d\times r}$ and an up-projection $W_{\text{up}} \in \mathbb{R}^{r\times d}$, where $r \ll d$ is the bottleneck dimension [Zhou *et al.*, 2024c]. Given an input feature x_{in}, the adapter output is injected via a residual connection to the MLP layer:

$$x_{\text{out}} = \text{MLP}(x_{\text{in}}) + W_{\text{up}}^\top \sigma(W_{\text{down}}^\top x_{\text{in}}), \tag{2}$$

where $\sigma(\cdot)$ denotes a non-linear activation function. Adapters are typically inserted into the MLP sublayer of each of the N_{blocks} transformer blocks.

To alleviate the significant parameter overhead introduced by maintaining separate adapters for every task, several recent methods adopt adapter merging strategies. The core idea is to maintain a single consolidated adapter model. As exemplified by ACMap [Fukuda *et al.*, 2025], after learning the adapter parameters θ_t for the new task, the method merges them into the global parameters $\bar{\theta}$ via an exponential moving average:

$$\begin{aligned}\theta_t^\star &= \arg\min_{\theta_t} \mathcal{L}_{ce}(\theta_t, D_t),\\ \bar{\theta}_t &= \alpha\bar{\theta}_{t-1} + (1-\alpha)\theta_t^\star.\end{aligned} \tag{3}$$

The overall performance of the final merged model is jointly determined by the compatibility of the adapter parameters and the selection of merging coefficients [Fukuda *et al.*, 2025; Qiu *et al.*, 2025].

4 Method

4.1 Overview

While merging-based methods have demonstrated competitive performance, they often suffer from significant catastrophic forgetting, as illustrated in Figure 1. A key reason for this limitation is that these approaches optimize each task-specific adapter θ_t solely to minimize the loss on the current task (cf. Figure 2(a)), without explicitly considering the performance of the merged model. As a result, naively aggregating incompatible adapter parameters leads to substantial accuracy degradation and exacerbates forgetting.

In contrast, we propose to optimize θ_t to directly maximize the performance of the merged model across all previously seen tasks. Formally, we frame this as a bi-level optimization problem (cf. Figure 2(b)):

$$\begin{aligned}\min_{\theta_t} \quad & \mathcal{L}_{\text{ce}}(\bar{\theta}_t, X_{1:t})\\ \text{s.t.} \quad & \bar{\theta}_t = \alpha^\star\bar{\theta}_{t-1} + (1-\alpha^\star)\theta_t,\\ & \alpha^\star = \arg\min_{\alpha}\ \phi(\alpha).\end{aligned} \tag{4}$$

Here, $\bar{\theta}_{t-1}$ denotes the merged adapter obtained after task $t-1$, and $\phi(\alpha)$ represents the objective function that determines the optimal merging coefficient (see Eq. (14) for details). In this formulation, the upper level optimizes the new task parameters θ_t to minimize the loss of the merged model, while the lower level determines the optimal merging coefficient $\alpha^\star$ that governs the aggregation of the previous and current models.

4.2 Upper Level Optimization

According to Eq. (4), the upper-level objective aims to optimize θ_t to minimize the cumulative loss of the merged model

IJCAI-ECAI 2026 PREPRINT

$\bar{\theta}_t$ over all observed data $X_{1:t}$. However, directly implementing it is infeasible due to the CIL constraint that the data from previous tasks (i.e., $X_{1:t-1}$) is no longer accessible.

To address this issue, we reformulate the optimization objective and derive the following decomposed form:

$$\begin{aligned} &\arg\min_{\theta_t} \mathcal{L}_{\text{ce}}(\bar{\theta}_t, X_{1:t}) \\ &= \arg\min_{\theta_t} \left(\mathcal{L}_{\text{ce}}(\bar{\theta}_t, X_t) + \mathcal{L}_{\text{ce}}(\bar{\theta}_t, X_{1:t-1})\right) \\ &= \arg\min_{\theta_t} \left(\mathcal{L}_{\text{ce}}(\theta_t, X_t) + \dot{\Delta}_t + \ddot{\Delta}_t\right), \end{aligned} \tag{5}$$

where the discrepancy terms are defined as follows:

$$\begin{aligned} \dot{\Delta}_t &= \mathcal{L}_{\text{ce}}(\bar{\theta}_t, X_t) - \mathcal{L}_{\text{ce}}(\theta_t, X_t), \\ \ddot{\Delta}_t &= \mathcal{L}_{\text{ce}}(\bar{\theta}_t, X_{1:t-1}) - \mathcal{L}_{\text{ce}}(\bar{\theta}_{t-1}, X_{1:t-1}). \end{aligned} \tag{6}$$

Note that $\mathcal{L}_{\text{ce}}(\bar{\theta}_{t-1}, X_{1:t-1})$ is constant with respect to θ_t and therefore does not influence the optimization. Consequently, the joint objective decomposes into three interpretable components: $\mathcal{L}_{\text{ce}}(\theta_t, X_t)$ optimizes the new task model θ_t on current data; $\dot{\Delta}_t$ encourages consistency between the merged model $\bar{\theta}_t$ and the new task model θ_t; and $\ddot{\Delta}_t$ promotes compatibility between the merged model $\bar{\theta}_t$ and the previous merged model $\bar{\theta}_{t-1}$. This decomposition transforms the intractable global objective into manageable sub-objectives, which can be approximately solved as described below.

Furthermore, following [Fukuda *et al.*, 2025], we decouple the optimization of the feature extractor and the classifier to better balance stability and plasticity. Specifically, we reformulate the discrepancy term $\dot{\Delta}_t$ in Eq. (6) as:

$$\begin{aligned} \dot{\Delta}_t &= \dot{\Delta}_t^C + \dot{\Delta}_t^F, \\ \dot{\Delta}_t^C &= \mathcal{L}_{\text{ce}}(\bar{\theta}_t^F, \bar{\theta}_t^C, X_t) - \mathcal{L}_{\text{ce}}(\bar{\theta}_t^F, \theta_t^C, X_t), \\ \dot{\Delta}_t^F &= \mathcal{L}_{\text{ce}}(\bar{\theta}_t^F, \theta_t^C, X_t) - \mathcal{L}_{\text{ce}}(\theta_t^F, \theta_t^C, X_t). \end{aligned} \tag{7}$$

Here, the superscripts F and C denote the feature extractor and classifier, respectively. A similar decomposition is applied to $\ddot{\Delta}_t$, yielding the sub-terms $\ddot{\Delta}_t^C$ and $\ddot{\Delta}_t^F$.

Feature extractor optimization. The discrepancies in the loss terms are primarily attributed to variations in the feature representations, given that the classifier is fixed:

$$\begin{aligned} \dot{\Delta}_t^F &\propto \left\| F(X_t; \bar{\theta}_t^F) - F(X_t; \theta_t^F) \right\|, \\ \ddot{\Delta}_t^F &\propto \left\| F(X_{1:t-1}; \bar{\theta}_t^F) - F(X_{1:t-1}; \bar{\theta}_{t-1}^F) \right\|. \end{aligned} \tag{8}$$

However, since the previous data $X_{1:t-1}$ is unavailable in Eq. (8), we approximate $\ddot{\Delta}_t^F$ using a first-order Taylor expansion around $\bar{\theta}_{t-1}^F$:

$$\ddot{\Delta}_t^F \approx \left\| \sum_{i=1}^{t-1} \nabla_{\theta_i^F} F(X_i; \theta_i^F) \left(\bar{\theta}_t^F - \bar{\theta}_{t-1}^F\right) \right\|, \tag{9}$$

where the accumulated gradients $\sum_{i=1}^{t-1} \nabla_{\theta_i^F} F(X_i; \theta_i^F)$ are maintained throughout training. Consequently, the overall optimization objective for the feature extractor is defined as:

$$\mathcal{L}_F = \dot{\Delta}_t^F + \ddot{\Delta}_t^F. \tag{10}$$

To further enhance discriminability between new and previously learned tasks in the merged model's feature space, we introduce a prototype-based loss:

$$\mathcal{L}_{\text{proto}} = \cos\left(F(X_t; \theta_t), P_{1:t-1}^{t-1}\right), \tag{11}$$

where $\cos(\cdot, \cdot)$ denotes cosine similarity and $P_{1:t-1}^{t-1}$ represents the set of class prototypes from all prior tasks. By penalizing high similarity between the feature representations of the current task and the prototypes of previous tasks, this loss encourages greater separation in the feature space, thereby improving inter-class discriminability.

Classifier optimization. Since MOBO employs a prototype-based classifier, optimization primarily involves updating these prototypes. For the discrepancy $\dot{\Delta}_t^C$ between the current and merged models, the availability of current task data enables direct computation and integration of the current task's prototype into the merged model. Conversely, for $\ddot{\Delta}_t^C$, where previous data is inaccessible, we approximate the prototypes of prior tasks in the merged model via prototype alignment. Additionally, to account for shifts in feature distribution introduced by new tasks, we incorporate an incremental covariance update. Specifically, the update mechanism is defined as:

$$\begin{aligned} P_{1:t-1}^{t} &= P_{1:t-1}^{t-1} + \left(P_t^t - P_t^{t-1}\right), \\ \hat{\Sigma}_t &= \frac{N_t \Sigma_t + N_{1:t-1} \hat{\Sigma}_{t-1}}{N_{1:t-1} + N_t}, \end{aligned} \tag{12}$$

where P_j^i denotes the prototype of task j within the i-th merged model, Σ_i is the task-specific feature covariance, and N_i is the sample size for task i. As illustrated in Figure 2 (c), these incremental updates to class prototypes and covariance matrices enable consistent modeling of class distributions across tasks. Leveraging these statistics, we employ a Mahalanobis-distance-based classifier during inference [Goswami *et al.*, 2023], thereby facilitating unified discrimination between old and new tasks.

Global loss. Building upon the preceding analysis, we formulate the global loss function as follows:

$$\mathcal{L}_{\text{global}} = \mathcal{L}_{\text{ce}}(\theta_t, X_t) + \lambda_1 \mathcal{L}_f + \lambda_2 \mathcal{L}_{\text{proto}}, \tag{13}$$

where λ_1 and λ_2 are hyperparameters balancing the contributions of each loss component. The global objective $\mathcal{L}_{\text{global}}$ consists of three terms: $\mathcal{L}_{\text{ce}}$, which ensures performance on the current task; $\mathcal{L}_f$, which optimizes the quality of the merged model; and $\mathcal{L}_{\text{proto}}$, which promotes class discriminability. Together, these components synergistically enhance generalization and model merging effectiveness.

4.3 Lower Level Optimization

As described in Eq. (4), the lower-level optimization seeks to determine the optimal merging coefficient α that enhances the merged model $\bar{\theta}_t$. Following the approach of [Qiu *et al.*, 2025], we compute this coefficient by minimizing the performance degradation caused by the merging process. Formally, this objective is expressed as:

$$\alpha^\star = \arg\min_{\alpha} \sum_{i=1}^{t} \mathcal{L}_i(\bar{\theta}_t) - \mathcal{L}_i(\theta_i), \tag{14}$$

IJCAI-ECAI 2026 PREPRINT

Algorithm 1 The optimization procedure of MOBO

Input: Dataset $\{(X_t, \mathcal{Y}_t)\}_{t=1}^{T}$, Iteration number N
Output: Merged parameters θ_T

if t = 1 **then**
  $\theta_1 \leftarrow \arg\min_{\theta_1} \mathcal{L}_{ce}(\theta_1, X_1)$
  $\bar{\theta}_1 = \theta_1$
end if
for $t = 2$ **to** T **do**
  for $iteration = 1$ **to** N **do**
    // The Upper Level, refer to Eq. (13)
    $\theta_t \leftarrow \arg\min_{\theta_t} \mathcal{L}_{\text{global}}(\theta_t, \bar{\theta}_t, X_t)$
    // The Lower Level, refer to Eq. (18)
    $q_i = (\bar{\theta}_{t-1} - \theta_t)^\top H_i(\theta_i)(\bar{\theta}_{t-1} - \theta_t)$
    $\alpha^\star = \frac{\sum_{i=1}^{t-1} q_i}{q_t + \sum_{i=1}^{t-1} q_i}$
    $\bar{\theta}_t = \alpha^\star \bar{\theta}_{t-1} + (1 - \alpha^\star)\theta_t$
  end for
end for
Return $\bar{\theta}_T$

where $\mathcal{L}_i(\bar{\theta}_t)$ denotes a shorthand for $\mathcal{L}_{ce}(\bar{\theta}_t, X_i)$.

To streamline computation and improve efficiency, we approximate $\mathcal{L}_i(\bar{\theta}_{t-1}) \approx \mathcal{L}_i(\theta_i)$ for all $i \leq t-1$. Thus, the objective function for the merging coefficient becomes:

$$\alpha^\star = \arg\min_{\alpha} \sum_{i=1}^{t-1} \Big(\mathcal{L}_i(\bar{\theta}_t) - \mathcal{L}_i(\bar{\theta}_{t-1})\Big) + \Big(\mathcal{L}_t(\bar{\theta}_t) - \mathcal{L}_t(\theta_t)\Big). \tag{15}$$

Next, we perform a second-order Taylor expansion of $\mathcal{L}_i(\bar{\theta}_t)$ around θ_i:

$$\mathcal{L}_i(\bar{\theta}_t) \approx \nabla\mathcal{L}_i(\theta_i)(\bar{\theta}_t - \theta_i) + \frac{1}{2}(\bar{\theta}_t - \theta_i)^\top H_i(\theta_i)(\bar{\theta}_t - \theta_i), \tag{16}$$

where $H_i(\theta_i)$ is the Hessian matrix of $\mathcal{L}_i$ evaluated at θ_i. Since θ_i is approximately a local minimizer of $\mathcal{L}_i$, the gradient term vanishes (i.e., $\nabla\mathcal{L}_i(\theta_i) \approx 0$). Under this assumption, Eq. (16) simplifies to:

$$\mathcal{L}_i(\bar{\theta}_t) \approx \frac{1}{2}(\bar{\theta}_t - \theta_i)^\top H_i(\theta_i)(\bar{\theta}_t - \theta_i). \tag{17}$$

Combining Eq.(15) and Eq.(17), we derive the analytical solution for the merging coefficient:

$$\begin{aligned} q_i &= (\bar{\theta}_{t-1} - \theta_t)^\top H_i(\theta_i)(\bar{\theta}_{t-1} - \theta_t). \\ \alpha^\star &= \frac{\sum_{i=1}^{t-1} q_i}{q_t + \sum_{i=1}^{t-1} q_i}. \end{aligned} \tag{18}$$

In practice, to reduce computational complexity, $H_i(\theta_i)$ is approximated and stored using the corresponding Fisher Information Matrix $F_i(\theta_i)$.

Finally, the overall MOBO procedure is summarized in Algorithm 1.

5 Experiments

5.1 Datasets for Evaluation

Datasets: We evaluate our method on five benchmark datasets: CIFAR-100 (CIFAR) [Krizhevsky and Hinton, 2009], CUB [Wah *et al.*, 2011], ImageNetR (IN-R) [Hendrycks *et al.*, 2021a], ImageNet-A (IN-A) [Hendrycks *et al.*, 2021b], and VTAB [Zhai *et al.*, 2019]. CIFAR-100 contains 100 classes, CUB, ImageNet-R, and ImageNet-A each contain 200 classes, and VTAB contains 50 classes. Following [Fukuda *et al.*, 2025], for all datasets except VTAB, the class order for each seed is randomized. For VTAB, the class order is fixed. We use the notation "B-m Inc-n" to divide these datasets into T tasks, where m is the initial number of classes and n is the number of classes added incrementally for each task.

Evaluation Metrics: Following the standard protocol in CIL [Rebuffi *et al.*, 2017], we use two evaluation metrics: the average accuracy $\bar{A}$ across all tasks and the final accuracy A_T (final accuracy) for the final task.

Baselines: We compare our method with several baselines and state-of-the-art methods. The baseline method is finetuning the pre-trained model for each task, referred to as Finetune. We also test finetuning only the adapter, referred to as Finetune Adapter. For comparison, we select CIL approaches using a pre-trained model: L2P [Wang *et al.*, 2022c], DualPrompt [Wang *et al.*, 2022b], CodaPrompt [Smith *et al.*, 2023], SimpleCIL [Zhou *et al.*, 2024a], APER [Zhou *et al.*, 2024a], EASE[Zhou *et al.*, 2024c], ACMap[Fukuda *et al.*, 2025], MOS[Sun *et al.*, 2025], and NSDCIL[Wu *et al.*, 2025]. Among these approaches, except for L2P , DualPrompt, and CodaPrompt, all other methods are adapter-based or lora-based solutions. Moreover, ACMap is our primary baseline for comparison, as it is the first work to consider an adapter-merging strategy for class-incremental learning.

Training Details: We follow the training configurations used in [Fukuda *et al.*, 2025]. For the pre-trained model, we use the ViT-B/16 model, pre-trained on ImageNet-21K [Ridnik *et al.*, 2021]. Adapter is optimized using SGD with cosine annealing for the learning rate scheduler. We tuned hyperparameters such as the learning rate, batch size, and training epochs for each dataset using the validation set, and report the final configurations in the supplementary materials.

5.2 Main Results

In this section, we compare our method with the state-of-the-art CIL approaches based on pre-trained models. Table 1 reports the average accuracy $\bar{A}$ and final accuracy A_T on the five benchmark datasets. In addition, Figure 3 illustrates the incremental performance trends of comparative methods. The experimental results demonstrate that our proposed method achieves strong performance in terms of both $\bar{A}$ and A_T, particularly on challenging datasets with significant domain shifts. For instance, on ImageNet-A, compared with ACMap, which also adopts a merging-based strategy, our approach achieves an additional performance gain of 8.69%. This substantial improvement highlights the benefit of incorporating global performance of the merged model into the training process, rather than relying solely on post hoc model merging. Moreover, our method attains a final accuracy of 62.34%, outperforming the suboptimal baseline NSDCIL by 3.16%, consistently validating its effectiveness.

We also compare with some traditional CIL methods in Table 2, including iCaRL[Rebuffi *et al.*, 2017], DER [Yan *et al.*,

IJCAI-ECAI 2026 PREPRINT

Method	CIFAR B0 Inc5		CUB B0 Inc5		IN-R B0 Inc5		IN-A B0 Inc10		VTAB B0 Inc10	
	$\bar{\mathcal{A}}$	$\mathcal{A}_T$	$\bar{\mathcal{A}}$	$\mathcal{A}_T$	$\bar{\mathcal{A}}$	$\mathcal{A}_T$	$\bar{\mathcal{A}}$	$\mathcal{A}_T$	$\bar{\mathcal{A}}$	$\mathcal{A}_T$
Finetune	38.90	20.17	23.20	11.79	21.61	10.79	21.60	10.96	34.95	21.25
Finetune Adapter	60.51	49.32	57.89	42.75	47.59	40.28	43.05	37.66	48.91	45.12
L2P[Wang *et al.*, 2022c]	85.94	79.93	56.71	49.06	66.53	59.22	47.16	38.48	77.11	77.10
DualPrompt[Wang *et al.*, 2022b]	87.87	81.15	58.34	50.29	63.31	55.22	52.56	42.68	83.36	81.23
CodaPrompt[Smith *et al.*, 2023]	89.11	81.96	59.89	50.76	64.42	55.08	48.51	36.47	83.90	83.02
SimpleCIL[Zhou *et al.*, 2024a]	87.7	81.26	92.45	86.73	62.58	54.55	60.65	48.91	85.99	84.38
APER[Zhou *et al.*, 2024a]	90.65	85.15	92.46	86.73	72.53	64.33	60.53	48.91	85.95	84.35
EASE[Zhou *et al.*, 2024c]	91.51	85.80	92.47	86.77	78.31	70.58	60.04	47.60	93.61	93.55
ACMap[Fukuda *et al.*, 2025]	92.04	87.81	91.89	86.17	77.31	70.49	64.30	53.65	91.21	87.56
MOS[Sun *et al.*, 2025]	93.30	89.25	93.15	88.89	76.79	69.53	64.38	53.19	92.62	92.79
NSDCIL[Wu *et al.*, 2025]	93.48	**90.51**	92.00	85.24	**80.40**	73.85	69.83	59.18	94.38	93.93
MOBO (Ours)	**94.13**	**90.51**	**93.74**	**89.44**	80.39	**74.45**	**71.37**	**62.34**	**94.82**	**94.79**

Table 1: Comparison of average and final performance across five datasets using the ViT-B/16-IN21K backbone. CIFAR refers to CIFAR-100, while IN-R and IN-A denote ImageNet-R and ImageNet-A, respectively. The best results are highlighted in **bold**, and the second-best results are indicated by an underline. All methods are implemented without the use of exemplars.

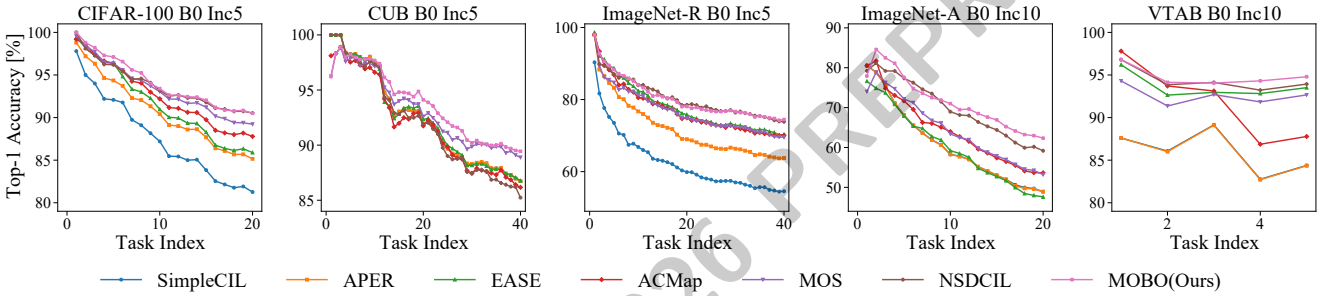

Figure 3: Accuracy curves across incremental tasks for adapter-based methods: SimpleCIL, APER, EASE, ACMap, MOS, and NSDCIL.

Method	Memory	CIFAR B0 Inc10		IN-R B0 Inc20	
		$\bar{\mathcal{A}}$	$\mathcal{A}_T$	$\bar{\mathcal{A}}$	$\mathcal{A}_T$
iCaRL	20/class	82.46	73.87	72.42	60.67
DER	20/class	86.04	77.93	80.48	74.32
FOSTER	20/class	89.87	84.91	**81.34**	74.48
MEMO	20/class	84.08	75.79	74.80	66.62
MOBO	-	**94.37**	**90.88**	81.08	**75.20**

Table 2: Comparison with traditional exemplar-based CIL methods, noting that our method operates without storing any exemplars.

Ablations	L_f	L_{proto}	α	$\bar{\mathcal{A}}$	$\mathcal{A}_T$
exp-I				64.30	53.65
exp-II	✓			70.72	61.62
exp-III	✓	✓		71.37	62.01
exp-IV	✓	✓	✓	71.37	62.34

Table 3: Ablation study analyzing the contribution of different components within MOBO on the ImageNet-A dataset.

2021], FOSTER [Wang *et al.*, 2022a] and MEMO [Zhou *et al.*, 2023]. To ensure a fair comparison , their backbones are also initialized with the pretrained ViT-B/16 model. As these methods require extra memory during incremental learning, we apply a fixed memory with 20 samples, following the iCaRL [Rebuffi *et al.*, 2017]. Notably, our method does not require any samples during either training or inference. Experimental results show that, on the CIFAR-100 dataset, our method achieves clear performance advantages over replay-based approaches. On the more challenging ImageNet-R dataset, our method further outperforms traditional replay-based CIL methods. Overall, even without sample storage, our method demonstrates strong competitiveness against traditional CIL approaches.

5.3 Ablation Study

As shown in Table 3, we conduct ablation studies on the ImageNet-A dataset to systematically evaluate the contribution of each component relative to the baseline. We adopt ACMap as the baseline method, which trains each task independently using the cross-entropy loss and merges model parameters via an average strategy. In exp-II, we introduce the feature-level loss $\mathcal{L}_f$, resulting in a performance improvement of 7.97%. In exp-III, we further incorporate

IJCAI-ECAI 2026 PREPRINT

Method	CIFAR B0 Inc2		CUB B0 Inc2		IN-R B0 Inc2		IN-A B0 Inc2		VTAB B0 Inc2	
	$\bar{\mathcal{A}}$	$\mathcal{A}_T$	$\bar{\mathcal{A}}$	$\mathcal{A}_T$	$\bar{\mathcal{A}}$	$\mathcal{A}_T$	$\bar{\mathcal{A}}$	$\mathcal{A}_T$	$\bar{\mathcal{A}}$	$\mathcal{A}_T$
ACMap[Fukuda *et al.*, 2025]	89.90	84.31	88.60	85.11	69.88	62.18	**61.45**	49.57	88.60	85.11
NSDCIL[Wu *et al.*, 2025]	91.97	85.12	85.72	69.89	65.13	51.88	49.35	38.78	90.84	89.64
MOBO (Ours)	**94.01**	**89.74**	**93.59**	**88.76**	**75.76**	**70.00**	61.38	**54.25**	**92.62**	**92.15**

Table 4: Comparison of average and final performance across five datasets in a 2-class incremental setting.

VTAB	EASE		MOS		ACMap		NSDCIL		MOBO	
	$\bar{\mathcal{A}}$	$\mathcal{A}_T$	$\bar{\mathcal{A}}$	$\mathcal{A}_T$	$\bar{\mathcal{A}}$	$\mathcal{A}_T$	$\bar{\mathcal{A}}$	$\mathcal{A}_T$	$\bar{\mathcal{A}}$	$\mathcal{A}_T$
B0 Inc10	93.61	93.55	92.79	92.62	91.21	87.56	94.38	93.93	**94.69**	**94.51**
B0 Inc5	88.68	88.84	92.93	92.28	90.83	85.90	93.19	92.56	**94.60**	**94.22**
B0 Inc2	85.34	86.24	**92.82**	91.01	88.60	85.11	90.84	89.64	92.62	**92.15**
B0 Inc1	87.36	84.44	**92.98**	90.03	87.36	84.38	90.35	89.37	92.58	**92.24**

Table 5: VTAB performance across varying task sequence lengths for different methods.

the prototype loss $\mathcal{L}_{proto}$, which achieves an additional gain of 0.39%. These results indicate that the global loss plays a critical role in enhancing the performance of the merged model. In particular, constraints imposed at the feature level improve compatibility between new and previously learned tasks, thereby effectively alleviating catastrophic forgetting. Finally, we replace the averaging strategy in ACMap with the Eq.(18), which results in a modest performance improvement. Overall, our method improves by approximately 8.69% over ACMap, further validating its effectiveness.

5.4 Long Tasks

In continual learning, models must continuously adapt to an evolving class distribution. Typically, the learning difficulty increases as the task sequence becomes longer. To evaluate the effectiveness and scalability of our proposed method under long task sequences, we conduct more challenging experiments, as reported in Table 4. In these experiments, each task in every dataset contains only two categories. For the CUB, ImageNet-A, and ImageNet-R datasets, this setting results in 100 consecutive tasks, representing a significantly more difficult learning scenario. We compare our method with ACMap, a representative merging-based approach, as well as NSDCIL, a suboptimal method that does not employ model merging. Compared with ACMap, our approach achieves superior performance across all datasets. Furthermore, relative to NSDCIL, our method achieves more pronounced improvements on the more challenging ImageNet-R and ImageNet-A datasets, with final accuracy gains of 18.12% and 15.47%, respectively. These results indicate that model merging is particularly advantageous in long task sequences and that the proposed merging-oriented bi-level optimization further enhances overall merged model performance.

We also evaluate the stability of our method as the task sequence length increases. Table 5 presents the performance of our method, compared with the baseline EASE, MOS, ACMap, and NSDCIL, across different task lengths. We conducted experiments with 5, 10, 25, and 50 tasks on the VTAB dataset. The results show that our method consistently outperforms the others on the final accuracy across all task sequence lengths, with its advantages becoming more pronounced as the task length increases. This trend indicates that our method remains effective as the number of tasks increases, demonstrating favorable scalability under long task sequences. In addition, our method maintains the highest performance stability throughout the task sequence, with the smallest overall degradation. In the B0 Inc1 scenario, our method attains a final accuracy of 92.24%, whereas the competing methods achieve a maximum of 90.03%. These results further underscore the effectiveness of our method in long-task scenarios.

6 Conclusion

In this paper, we proposed a Merging-Oriented Bi-level Optimization framework for class incremental learning (MOBO). First, we designed a global loss that treats the overall performance of the merged model as the primary training objective. Second, we adopted a bi-level optimization framework that alternately optimizes the new task parameters and the merged model parameters. The upper level updates the new task parameters using the global loss, guiding them toward a solution space that is favorable for subsequent merging. The lower level updates the merged model parameters by solving for the optimal merging coefficients. This approach dynamically balances new and old knowledge from a global perspective, effectively mitigating catastrophic forgetting and improving the final performance of the merged model. Experimental results demonstrate that our method achieves state-of-the-art performance across all datasets and yields significant improvements on long task sequences. However, while we achieved progress in solving for merge coefficients compared to prior methods, the gains remain limited. In future work, we will focus on further optimizing the selection of merge coefficients to enhance model performance.

IJCAI-ECAI 2026 PREPRINT

Acknowledgments

This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 2022JBQY007, in part by the National Natural Science Foundation of China under Grant 62276019, 62306028, in part by the Beijing Natural Science Foundation under Grant L231019.

References

[Aljundi *et al.*, 2018] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision*, pages 139–154, 2018.

[Aljundi *et al.*, 2019a] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11254–11263, 2019.

[Aljundi *et al.*, 2019b] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

[Belouadah and Popescu, 2019] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 583–592, 2019.

[Chaudhry *et al.*, 2019] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *Proceedings of the International Conference on Learning Representations*, 2019.

[Dhar *et al.*, 2019] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5138–5146, 2019.

[Douillard *et al.*, 2020] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Proceedings of the European Conference on Computer Vision*, pages 86–102. Springer, 2020.

[Fukuda *et al.*, 2025] Takuma Fukuda, Hiroshi Kera, and Kazuhiko Kawamoto. Adapter merging with centroid prototype mapping for scalable class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4884–4893, 2025.

[Goswami *et al.*, 2023] Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost Van De Weijer. Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. *Advances in Neural Information Processing Systems*, 36:6582–6595, 2023.

[Han *et al.*, 2024] Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. Parameter-efficient fine-tuning for large models: A comprehensive survey. *Transactions on Machine Learning Research*, 2024.

[Hendrycks *et al.*, 2021a] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021.

[Hendrycks *et al.*, 2021b] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021.

[Kirkpatrick *et al.*, 2017] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

[Krizhevsky and Hinton, 2009] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[Li and Hoiem, 2017] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017.

[Liu *et al.*, 2020] Yaoyao Liu, Yuting Su, An-An Liu, Bernt Schiele, and Qianru 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.

[Pham *et al.*, 2022] Quang Pham, Chenghao Liu, and Steven Hoi. Continual normalization: Rethinking batch normalization for online continual learning. In *Proceedings of the International Conference on Learning Representations*, 2022.

[Pian *et al.*, 2023] Weiguo Pian, Shentong Mo, Yunhui Guo, and Yapeng Tian. Audio-visual class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7799–7811, 2023.

[Qiu *et al.*, 2025] Haomiao Qiu, Miao Zhang, Ziyue Qiao, and Liqiang Nie. Train with perturbation, infer after merging: A two-stage framework for continual learning. *Advances in Neural Information Processing Systems*, 2025.

[Ratcliff, 1990] Roger Ratcliff. Connectionist models of recognition memory: Constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285, 1990.

[Rebuffi *et al.*, 2017] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017.

IJCAI-ECAI 2026 PREPRINT

[Ridnik *et al.*, 2021] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021.

[Smith *et al.*, 2023] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11909–11919, 2023.

[Sun *et al.*, 2025] Hai-Long Sun, Da-Wei Zhou, Hanbin Zhao, Le Gan, De-Chuan Zhan, and Han-Jia Ye. Mos: Model surgery for pre-trained model-based class-incremental learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 20699–20707, 2025.

[Thrun, 1995] Sebastian Thrun. Is learning the n-th thing any easier than learning the first? *Advances in Neural Information Processing Systems*, 8, 1995.

[Wah *et al.*, 2011] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[Wang *et al.*, 2022a] Fu-Yun Wang, Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Foster: Feature boosting and compression for class-incremental learning. In *Proceedings of the European Conference on Computer Vision*, pages 398–414. Springer, 2022.

[Wang *et al.*, 2022b] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *Proceedings of the European Conference on Computer Vision*, pages 631–648. Springer, 2022.

[Wang *et al.*, 2022c] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022.

[Wu *et al.*, 2025] Fangwen Wu, Lechao Cheng, Shengeng Tang, Xiaofeng Zhu, Chaowei Fang, Dingwen Zhang, and Meng Wang. Navigating semantic drift in task-agnostic class-incremental learning. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

[Yan *et al.*, 2021] Shipeng Yan, Jiangwei Xie, and Xuming 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.

[Zhai *et al.*, 2019] Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

[Zhou *et al.*, 2023] Da-Wei Zhou, Qi-Wei Wang, Han-Jia Ye, and De-Chuan Zhan. A model or 603 exemplars: Towards memory-efficient class-incremental learning. In *Proceedings of the International Conference on Learning Representations*, 2023.

[Zhou *et al.*, 2024a] Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *International Journal of Computer Vision*, 133(3):1012–1032, 2024.

[Zhou *et al.*, 2024b] Da-Wei Zhou, Hai-Long Sun, Jingyi Ning, Han-Jia Ye, and De-Chuan Zhan. Continual learning with pre-trained models: A survey. In *International Joint Conference on Artificial Intelligence*, pages 8363–8371, 2024.

[Zhou *et al.*, 2024c] Da-Wei Zhou, Hai-Long Sun, Han-Jia Ye, and De-Chuan Zhan. Expandable subspace ensemble for pre-trained model-based class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23554–23564, 2024.

IJCAI-ECAI 2026 PREPRINT