BOFA: Bridge-Layer Orthogonal Low-Rank Fusion for CLIP-based Class-Incremental Learning

Lan Li[1,2], Tao Hu[1,2], Da-Wei Zhou[1,2*], Han-Jia Ye[1,2] De-Chuan Zhan[1,2]
[1]School of Artificial Intelligence, Nanjing University, China
[2]National Key Laboratory for Novel Software Technology, Nanjing University, China
{lil,hut,zhoudw,yehj,zhandc}@lamda.nju.edu.cn

Abstract

Class-Incremental Learning (CIL) aims to continually learn new categories without forgetting previously acquired knowledge. Vision-language models such as CLIP offer strong transferable representations via multi-modal supervision, making them promising for CIL. However, applying CLIP to CIL poses two major challenges: (1) adapting to downstream tasks often requires additional learnable modules, increasing model complexity and susceptibility to forgetting; and (2) while multi-modal representations offer complementary strengths, existing methods have yet to fully realize their potential in effectively integrating visual and textual modalities. To address these issues, we propose BOFA (Bridge-layer Orthogonal Fusion for Adaptation), a novel framework for CIL. BOFA confines all model adaptation exclusively to CLIP's existing cross-modal bridge-layer, thereby adding no extra parameters or inference cost. To prevent forgetting within this layer, it leverages Orthogonal Low-Rank Fusion, a mechanism that constrains parameter updates to a low-rank "safe subspace" mathematically constructed to be orthogonal to past task features. This ensures stable knowledge accumulation without data replay. Furthermore, BOFA employs a cross-modal hybrid prototype that synergizes stable textual prototypes with visual counterparts derived from our stably adapted bridge-layer, enhancing classification performance. Extensive experiments on standard benchmarks show that BOFA achieves superior accuracy and efficiency compared to existing methods.

Extended version —
https://aaai.org/example/extended-version

Introduction

In the real world, data is often encountered in a streaming fashion, where new categories arrive sequentially (Aggarwal 2018; Li et al. 2024). Class-Incremental Learning (CIL) (Rebuffi et al. 2017) addresses this challenge by enabling models to learn novel classes without forgetting previously acquired knowledge. However, continual learning systems are prone to catastrophic forgetting (French 1999; French and Ferrara 1999), a phenomenon where learning new tasks interferes with performance on earlier ones. This problem is particularly pronounced when old data is unavailable due to privacy or storage constraints.

Recent advances in vision-language models (VLMs) (Jia et al. 2021; Yang et al. 2023; Yu et al. 2024a; Yuan et al. 2021; Li et al. 2023), such as CLIP (Radford et al. 2021), have opened new possibilities for CIL. By aligning visual and textual modalities in a shared embedding space, VLMs provide rich, transferable representations that can generalize to unseen tasks. This makes them a compelling foundation for exemplar-free CIL.

A prevailing strategy is to freeze the large backbones of CLIP to preserve general knowledge and only train a small set of additional modules, such as adapters or prompt layers, for new tasks (Zhou et al. 2025c; Huang et al. 2024). While this approach maintains a stable high-level feature space, it introduces a new set of critical challenges. First, catastrophic forgetting persists within the trainable modules themselves. As adapters are sequentially fine-tuned on new tasks to perform cross-modal alignment, they overwrite knowledge crucial for old tasks, leading to a degradation in performance. Second, these additional parameters, however lightweight, result in an incremental inference cost, which poses a practical limitation for deployment in resource-constrained scenarios. Third, the training process for these modules is often suboptimal. They are typically trained via contrastive learning against textual prototypes derived from hand-crafted prompts. These generic prompts often lack the specificity needed to distinguish fine-grained visual concepts (Wang et al. 2022b, 2023b; Zhou et al. 2025c), thereby limiting the model's ultimate discriminative capability.

To address these challenges, we propose BOFA (Bridge-layer Orthogonal Fusion for Adaptation), a unified framework for efficient and continual adaptation of CLIP in CIL. Our approach first tackles the dual issues of inference overhead and catastrophic forgetting through a coupled strategy. The foundation of our method is to confine all adaptation exclusively to CLIP's existing cross-modal bridge-layer, thereby eliminating the need for external modules and their associated inference costs. This design, however, concentrates the risk of catastrophic forgetting within this single component. To counteract this, we introduce our primary technical contribution: Orthogonal Low-Rank Fusion. This mechanism constrains parameter updates to an "Orthogonal Safe Subspace", a low-rank manifold mathematically con-

*Corresponding author.

structed to be approximately orthogonal to the feature space of previously learned tasks. By projecting updates into these non-interfering directions, BOFA ensures stable knowledge accumulation within the bridge-layer, effectively mitigating catastrophic forgetting without requiring data replay. Building upon this stable adaptation, we introduce cross-modal hybrid Prototypes to enhance classification. Our mechanism fuses static textual prototypes with dynamically refined visual prototypes, which are generated by our stably adapted bridge-layer. This synergy yields a robust and highly discriminative classifier, surpassing unimodal baselines.

In summary, our key contributions include: (1) We introduce orthogonal low-rank fusion, a novel and efficient adaptation framework for CIL. By constraining updates within CLIP's existing bridge-layer to an orthogonal safe subspace, our method mitigates catastrophic forgetting without incurring any additional parameters or inference costs. (2) We design a cross-modal hybrid prototype that leverages our stable adaptation to fuse textual and visual prototypes, resulting in a more discriminative classifier. (3) The proposed BOFA achieves SOTA performance on multiple CIL benchmarks, demonstrating superior accuracy and efficiency.

Related Work

Class-Incremental Learning (CIL). CIL aims to enable models to incrementally learn new classes without forgetting previously acquired knowledge (Masana et al. 2022; De Lange et al. 2021). Traditional CIL approaches can be broadly categorized into several groups. Distillation-based methods mitigate forgetting by aligning outputs between the current and previous models (Hinton, Vinyals, and Dean 2015; Douillard et al. 2020), through logit-level (Rebuffi et al. 2017; Li and Hoiem 2016), feature-level (Lu, Wang, and Deng 2022; Park, Kang, and Han 2021), or group-level alignment (Gao et al. 2022; Dong et al. 2021). Replay-based methods explicitly preserve past knowledge by storing and revisiting samples from previous tasks (Luo et al. 2023; Chaudhry et al. 2018). Regularization-based approaches estimate parameter importance and penalize updates to critical weights (Aljundi et al. 2018; Aljundi, Kelchtermans, and Tuytelaars 2019). Bias rectification methods focus on correcting prediction or classifier bias accumulated during incremental training (Shi et al. 2022; Zhao et al. 2020). Model expansion methods dynamically increase model capacity by expanding neurons (Yoon et al. 2018; Xu and Zhu 2018), backbones (Zhou et al. 2023; Wang et al. 2023a; Zheng et al. 2025), or lightweight components (Douillard et al. 2022) to accommodate new knowledge.

Pre-Trained Model-Based CIL. Leveraging pre-trained models has emerged as a promising direction for CIL (Zhou et al. 2025b; Qi et al. 2025; Sun et al. 2025), as they offer generalizable representations that can accelerate learning and improve performance. A prevalent strategy is to freeze the pre-trained backbone and introduce lightweight, learnable modules such as prompts (Wang et al. 2022a,b; Smith et al. 2023; Wang, Huang, and Hong 2022; Zhou et al. 2022) and adapters (Yu et al. 2024b; Chen et al. 2022). L2P (Wang et al. 2022b) and DualPrompt (Wang et al. 2022a) utilize learnable prompt pools with selection mechanisms tailored for pre-trained vision transformers (Jia et al. 2022). Further extensions explore more sophisticated prompt composition using attention (Smith et al. 2023) or generative networks (Jung et al. 2023). Another line of work builds classifiers directly on top of pre-trained embeddings, matching class prototypes to feature representations (Zhou et al. 2025a; McDonnell et al. 2023; Snell, Swersky, and Zemel 2017). For multi-modal settings with pre-trained CLIP, several approaches design cross-modal prompt tuning schemes to enhance alignment (Wang et al. 2023b; Wang, Huang, and Hong 2022). MOE-Adapter (Yu et al. 2024b) introduces mixture-of-experts mechanisms (Masoudnia and Ebrahimpour 2014) for adaptive module selection, while PROOF (Zhou et al. 2025c) extends CLIP's representational capacity with task-specific projection heads. RAPF (Huang et al. 2024) further decomposes parameter updates into modular components, enabling adaptive adapter fusion to balance plasticity and stability.

Preliminaries

This section introduces the exemplar-free CIL setup and outlines how CLIP can be adapted to incremental learning.

Class-Incremental Learning

Class-incremental learning (CIL) aims to incrementally construct a classifier that is capable of recognizing all classes encountered over a sequence of tasks in a data stream setting (Rebuffi et al. 2017). We denote the sequence of training sets as $\{\mathcal{D}^1, \mathcal{D}^2, \ldots, \mathcal{D}^T\}$, where the dataset for task t is given by $\mathcal{D}^t = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_t}$. Given this setup, each dataset consists of input-label pairs $(\mathbf{x}_i, y_i)$, where each input $\mathbf{x}_i \in \mathbb{R}^D$ is associated with a label y_i drawn from the task-specific label set Y_t, with $Y_t \cap Y_{t'} = \varnothing$ for $t \neq t'$.

This work focuses on the **exemplar-free** CIL setting, where no historical instances can be stored for rehearsal (Zhu et al. 2021; Wang et al. 2022b,a). Consequently, during the t-th incremental step, the model has access only to the current task data $\mathcal{D}^t$. The objective is to train a classifier f over the union of all seen label sets $\mathcal{Y}_t = \bigcup_{t=1}^{T} Y_t$ that minimizes the expected misclassification risk:

$$f^* = \operatorname*{arg\,min}_{f \in \mathcal{H}} \mathbb{E}_{(\mathbf{x},y)\sim \bigcup_{t=1}^{T} \mathcal{D}^t} \left[\mathbb{I}(y \neq f(\mathbf{x})) \right],$$

where $\mathcal{H}$ is the hypothesis space, $\mathbb{I}(\cdot)$ is the indicator function, and $\mathcal{D}^t$ denotes the data of task t.

CLIP-based CIL

We build upon CLIP (Radford et al. 2021), a VLM pre-trained on image–text pairs, as the foundation for our CIL framework, following prior work (Zhou et al. 2025c; Yu et al. 2024b; Huang et al. 2024).

CLIP architecture. CLIP employs two modality-specific encoders to project images and texts into a shared d-dimensional embedding space: a text encoder and an image encoder. The image encoder is typically decomposed as $g_i = g_2 \circ g_1$, where g_1 is a visual backbone (e.g., ViT) that extracts raw visual features $\mathbf{x}_o \in \mathbb{R}^{d_o}$. The subsequent layer, g_2, is a linear projection that maps these features into the joint embedding space. We refer to g_2, parameterized by its weight matrix $\mathbf{W} \in \mathbb{R}^{d_o \times d}$ as the **cross-modal bridge-layer**. The final image embedding is thus computed as

$$\mathbf{x}_o = g_1(\mathbf{x}), \quad \mathbf{z} = g_2(\mathbf{x}_o) = \mathbf{x}_o \mathbf{W},$$

Zero-Shot Classification with Textual Prototypes. A key capability of CLIP is its zero-shot classification performance. For a given set of classes, a classifier can be constructed on-the-fly from their names. Each class name is embedded into a textual prompt, such as "a photo of a `[CLASS]`", and then encoded by g_t to form a **textual prototype**. An input image $\mathbf{x}$ is classified by computing the cosine similarity between its visual embedding $\mathbf{z}_i$ and each textual prototype $\mathbf{z}_t^c$:

$$P(y = c \mid \mathbf{x}) = \frac{\exp(\cos(\mathbf{z}_i, \mathbf{z}_t^c)/\tau)}{\sum_{c'} \exp(\cos(\mathbf{z}_i, \mathbf{z}_t^{c'})/\tau)}, \tag{1}$$

where τ is a temperature parameter.

Adapting CLIP for Incremental Learning. While the zero-shot classifier provides a strong baseline, the model's performance can be further improved by fine-tuning on downstream training data, typically using the cross-entropy loss on the probabilities in Eq. (1). In the context of CIL, a prevalent adaptation strategy is to freeze the large, pre-trained backbones (g_i and g_t) to preserve their general knowledge. To learn task-specific information, a lightweight, trainable module, such as an adapter, is introduced after the image encoder. This module is then sequentially updated as new tasks arrive.

Discussion. Although CLIP provides a promising foundation for CIL, its adaptation faces several key challenges. First, the dominant approach of adding an external adapter, while protecting the model's backbone, merely shifts the problem to the adapter. As this compact module is sequentially updated, it still struggles to retain knowledge of previous tasks, leading to significant performance degradation. This also increases the model's parameter count and inference latency. Second, the reliance on textual prototypes makes the classifier's quality highly sensitive to prompt engineering and often results in suboptimal alignment with data-specific visual features. Therefore, an effective CLIP-based CIL approach should mitigate forgetting without introducing additional parameters and robustly leverage cross-modal information.

BOFA: Bridge-layer Orthogonal Fusion for Adaptation

To tackle the challenges of catastrophic forgetting and suboptimal modality alignment discussed above, we design our BOFA around three key components: (1) lightweight fine-tuning of CLIP's bridge-layer, (2) orthogonal low-rank fusion to preserve previous knowledge, and (3) cross-modal hybrid prototypes to enhance classification performance.

Fine-tuning the Cross-Modal Bridge-Layer

Most prior CLIP-based CIL methods introduce external trainable modules (e.g., adapters or prompt layers) to bridge the gap between new visual and textual concepts (Zhou et al. 2025c; Huang et al. 2024). We propose a streamlined alternative that **leverages CLIP's existing parameters without introducing external modules or increasing the model size**. Specifically, we fine-tune only the projection layer g_2, which serves as the cross-modal bridge-layer that maps the high-dimensional visual features $\mathbf{x}_o = g_1(\mathbf{x})$ extracted by the frozen visual backbone g_1 into the shared embedding space. By updating only this layer while keeping both the visual backbone g_1 and the text encoder g_t frozen, we maintain the original CLIP architecture without introducing additional parameters. This adaptation leverages the semantic richness and generality of the high-dimensional visual feature space $\mathbb{R}^{d_o}$, prior to projection, enabling flexible and efficient learning of new tasks. Moreover, since only a lightweight layer is fine-tuned, our method does not incur any additional inference cost, making it an efficient and practical solution for CIL with CLIP.

Orthogonal Low-Rank Fusion

While fine-tuning the cross-modal bridge-layer enables efficient adaptation to new tasks, preserving past knowledge remains a critical challenge in CIL. Naive fine-tuning disrupts previously learned image-text alignments, causing catastrophic forgetting. To address this issue, we propose orthogonal low-rank fusion, a framework that augments fine-tuning with a principled, low-dimensional constraint on weight updates to minimize interference with past tasks. By leveraging the approximate null space of previous task features, our method effectively preserves past knowledge while adapting to new tasks. Furthermore, we use Low-Rank Adaptation (LoRA) (Hu et al. 2022) to efficiently implement updates in the approximate null space.

Forgetting Analysis. Let $\mathbf{W}_{\text{old}} = \mathbf{W}_0 + \Delta\mathbf{W}_{\text{old}}$ be the weights of the bridge-layer after training on a sequence of past tasks, where $\mathbf{W}_0$ is the pretrained weight and $\Delta\mathbf{W}_{\text{old}}$ is the fused parameter update of past tasks. To adapt to a new task, fine-tuning produces a parameter update $\Delta\mathbf{W}_{\text{new}}$ of new task. After adaptation, the fused weight matrix becomes $\mathbf{W}_{\text{new}} = \mathbf{W}_{\text{old}} + \Delta\mathbf{W}_{\text{new}}$. For the feature matrix $\mathbf{X}_{\text{old}}$ collected from previous tasks (each row representing the visual feature $\mathbf{x}_{\text{o}}$ of a previous sample), the projected embeddings are perturbed from $\mathbf{X}_{\text{old}}\mathbf{W}_{\text{old}}$ to:

$$\mathbf{X}_{\text{old}}\mathbf{W}_{\text{new}} = \mathbf{X}_{\text{old}}\mathbf{W}_0 + \mathbf{X}_{\text{old}}\Delta\mathbf{W}_{\text{old}} + \underbrace{\mathbf{X}_{\text{old}}\Delta\mathbf{W}_{\text{new}}}_{\text{Interference Term}}.$$

The interference term is the primary source of forgetting. If the cumulative effect of this interference across all past features is significant, the representations of previous tasks are substantially degraded. To mitigate this, the core idea is to constrain the update $\Delta\mathbf{W}_{\text{new}}$ such that its effect on the feature space of previous tasks, represented by the feature matrix $\mathbf{x}_{o,\text{old}}$, is minimized. Ideally, we seek to satisfy:

$$\mathbf{X}_{\text{old}}\Delta\mathbf{W}_{\text{new}} \approx \mathbf{0}. \tag{2}$$

This condition implies that each column of the update matrix $\Delta\mathbf{W}_{\text{new}}$ should belong to the **null space** of $\mathbf{X}_{\text{old}}$, thereby preserving knowledge of past tasks while learning new ones.

Constructing an Orthogonal Safe Subspace. In practice, the high-dimensional feature matrix $\mathbf{x}_{o,\text{old}}$, comprising diverse data from multiple tasks, is typically full-rank. Consequently, its exact null space is trivial (containing only the zero vector), which renders the strict constraint in Eq. (2) impractical. To address the challenge of catastrophic forgetting, we relax the notion of an exact "null space" into an approximate space. Specifically, we define an approximate null space as any subspace into which the projection of previous task features has low magnitude. Constraining weight updates to lie within such a subspace effectively minimizes interference with representations from previous tasks.

Let $\mathbf{P} \in \mathbb{R}^{d_o \times k}$ represent an orthonormal basis matrix such that $\mathbf{P}^\top \mathbf{P} = \mathbf{I}_k$. We measure the interference induced by this basis on the old features using the Frobenius norm:

$$\mathcal{I}(\mathbf{P}) = \left\|\mathbf{X}_{\text{old}}\mathbf{P}\right\|_F^2,$$

and define the optimal k-dimensional approximate null space as:

$$\mathbf{P}^* = \arg\min_{\mathbf{P}^\top\mathbf{P}=\mathbf{I}_k} \left\|\mathbf{X}_{\text{old}}\mathbf{P}\right\|_F^2. \tag{3}$$

Proposition 1 *The cumulative scatter matrix of past features is defined as* $\mathbf{S}_{old} = \mathbf{X}_{old}^\top \mathbf{X}_{old}$*, which can be decomposed as:*

$$\mathbf{S}_{old} = \mathbf{U}\,\text{diag}(\lambda_1, \ldots, \lambda_{d_o})\mathbf{U}^\top, \quad \lambda_1 \geq \cdots \geq \lambda_{d_o}.$$

The optimal solution to Eq. (3) *is given by:*

$$\mathbf{P}^* = \left[\mathbf{u}_{d_o-k+1}, \ldots, \mathbf{u}_{d_o}\right],$$

which is the subspace spanned by the eigenvectors of $\mathbf{S}_{old}$ *associated with its* k *smallest eigenvalues.*

We term the subspace spanned by $\mathbf{P}^*$ the Orthogonal Safe Subspace (OSS). It represents the k-dimensional subspace that minimizes the projection of past features, thus identifying the directions of least interference. To mitigate catastrophic forgetting, we constrain the parameter update $\Delta\mathbf{W}_{\text{new}}$ to lie within this subspace. We subsequently employ a modified LoRA scheme to implement this constraint efficiently, as detailed below. A full proof of Proposition 1 is provided in Appendix A.2. It is worth noting that in CIL scenarios, maintaining the OSS does not require storing all previous task features. The cumulative scatter matrix is incrementally updated as:

$$\mathbf{S}_{\text{new}} = \mathbf{S}_{\text{old}} + \mathbf{X}_{\text{new}}^\top \mathbf{X}_{\text{new}}.$$

The updated OSS is then efficiently computed from the k smallest eigenvectors of $\mathbf{S}_{\text{new}}$, eliminating the need to store past data.

LoRA in the Orthogonal Safe Subspace. To implement the update constraint efficiently, we adapt the LoRA framework. Standard LoRA approximates the weight update as $\Delta\mathbf{W} = \mathbf{AB}$, where $\mathbf{A} \in \mathbb{R}^{d_o \times k}$ and $\mathbf{B} \in \mathbb{R}^{k \times d}$ are trainable low-rank matrices. Our goal is to enforce that the row space of $\Delta\mathbf{W}$ lies within the OSS. Since the rows of $\Delta\mathbf{W}$ are linear combinations of the rows of $\mathbf{A}$, we fix the low-rank matrix $\mathbf{A}$ to be the OSS basis, i.e., we set $\mathbf{A} = \mathbf{P}^*$, where $\mathbf{P}^*$ is the orthonormal basis spanning the approximate null space as defined above. With this parameterization, the weight update is given by $\Delta\mathbf{W} = \mathbf{P}^*\mathbf{B}$, where only $\mathbf{B}$ remains trainable.

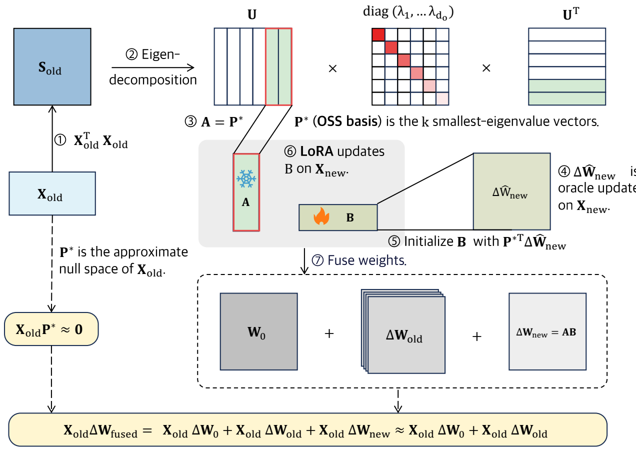


Figure 1: Overview of Orthogonal Low-Rank Fusion, where an OSS $\mathbf{P}^*$ is constructed from past task features to constrain the low-rank update for a new task, thereby minimizing interference with prior knowledge.

However, when $\mathbf{A}$ is frozen to the orthogonal safe subspace basis, simply initializing $\mathbf{B}$ to zero can lead to optimization difficulties, as the initial optimization direction may not align with the new task's objectives within the OSS. To address this, we use a data-driven initialization for $\mathbf{B}$. We first obtain a temporary "oracle" update $\Delta\tilde{\mathbf{W}}_{\text{new}}$ by briefly fine-tuning the entire bridge layer on the new task data. Our goal is to initialize $\mathbf{B}$ by minimizing the discrepancy $\|\Delta\tilde{\mathbf{W}}_{\text{new}} - \mathbf{AB}\|_F^2$ with $\mathbf{A}$ frozen. When $\mathbf{A}$ is set to the basis $\mathbf{P}^*$, this yields the closed-form solution

$$\mathbf{B}_0 = \mathbf{P}^{*\top}\Delta\tilde{\mathbf{W}}_{\text{new}}.$$

This initialization provides a safe and task-adaptive update, accelerating convergence and improving optimization stability.

Figure 1 provides an overview of the complete orthogonal low-rank fusion pipeline, including the construction of the orthogonal safe subspace and its LoRA-based implementation. After the current task is learned, the bridge-layer parameters are updated as

$$\mathbf{W}_{\text{fused}} = \mathbf{W}_0 + \Delta\mathbf{W}_{\text{old}} + \Delta\mathbf{W}_{\text{new}}.$$

Learning with Cross-Modal Hybrid Prototypes

As established in our preliminaries, relying solely on textual prototypes can limit classification performance. To address this, we introduce a cross-modal hybrid classifier that fuses general semantic knowledge from text with data-driven visual characteristics.

Static Hybrid Prototype Construction. For each class c, we first construct a static hybrid prototype $\mathbf{p}_c$ by linearly interpolating between its textual prototype $\mathbf{z}_t^c$ and its visual prototype $\mathbf{z}_i^c$:

$$\mathbf{p}_c = (1-\lambda)\,\mathbf{z}_t^c + \lambda\,\mathbf{z}_i^c. \tag{4}$$

The textual prototype $\mathbf{z}_t^c$ is the pre-computed embedding from the frozen text encoder g_t. The visual prototype $\mathbf{z}_i^c$ is obtained by applying the bridge-layer g_2 to the mean high-dimensional feature $\bar{\mathbf{x}}_o^c$ of class c. The balancing coefficient λ is determined via a grid search on the first task's training set and is then fixed for all subsequent tasks.

Dynamic Prototype Refinement. A key challenge during incremental learning is that the bridge-layer g_2 is continuously updated, causing its output feature space to shift. To mitigate this, we introduce a dynamic refinement process. During the training of each task, we continuously update the visual prototype $\mathbf{z}_i^c$ of each seen class using an Exponential Moving Average (EMA). This ensures that the prototypes stay current with the model's learned representations. At the conclusion of the entire incremental sequence, we perform a final refinement step by recomputing all visual prototypes using the final adapted bridge-layer $\mathbf{W}_{\text{fused}}$ before fusing them for inference.

Hierarchical Inference Strategy. To further boost performance at inference time, we employ a two-stage hierarchical classification process. First, for an input image, we use a set of lightweight, task-specific auxiliary classifiers (trained on the high-dimensional features $\mathbf{x}_o$) to efficiently pre-select a small subset of the most likely candidate classes. In the second stage, the primary model performs the final, more precise classification using the refined hybrid prototypes $\{\mathbf{p}_c\}$, but only within this reduced candidate set. This hierarchical approach effectively narrows the search space, reducing ambiguity and enhancing overall accuracy.

Discussion of BOFA

In summary, BOFA's methodology rests on three synergistic components. First, we perform lightweight fine-tuning exclusively on CLIP's existing cross-modal bridge-layer, avoiding additional parameters. Second, to protect past knowledge, we apply proposed orthogonal low-rank fusion to constrain this bridge-layer's updates, ensuring they lie within an Orthogonal Safe Subspace that minimizes interference. Finally, we enhance classification accuracy using data-driven cross-modal hybrid prototypes. This integrated design is highly memory-efficient. Its primary storage cost stems from maintaining the cumulative scatter matrix ($\mathbb{R}^{d_o \times d_o}$), as well as storing both the mean high-dimensional feature for each class and the auxiliary classifiers (each requiring $|\mathcal{Y}| \cdot d_o$ in total). This is a significant improvement over methods requiring data replay like PROOF (Zhou et al. 2025c) or large per-class covariance matrices (approx. $|\mathcal{Y}| \cdot d^2$) like RAPF (Huang et al. 2024). Crucially, since no external modules are added, BOFA preserves the original inference cost of CLIP, making it a practical and efficient solution for class-incremental learning. A detailed cost analysis and the pseudocode can be found in the Appendix.

Experiments

Implementation Details

Dataset: Following prior works (Zhou et al. 2025c, 2022; Wang et al. 2022b), we evaluate performance on nine benchmark datasets that exhibit significant domain shift from CLIP's pre-training data. These include: *CIFAR100*(Krizhevsky 2009), *CUB200*(Wah et al. 2011), *ObjectNet*(Barbu et al. 2019), *ImageNet-R*(Hendrycks et al. 2021), *FGVCAircraft*(Maji et al. 2013), *StanfordCars*(Krause et al. 2013), *Food101*(Bossard, Guillaumin, and Van Gool 2014), *SUN397*(Xiao et al. 2010), and *UCF101*(Soomro, Zamir, and Shah 2012). To facilitate class-incremental splits, we follow the sampling strategy in(Zhou et al. 2025c): selecting 100 classes from CIFAR100, Aircraft, Cars, Food101, and UCF101; 200 classes from CUB200, ObjectNet, and ImageNet-R; and 300 classes from SUN397. Detailed dataset statistics and splits are provided in the supplementary material.

Dataset split: Following the convention in (Rebuffi et al. 2017; Wang et al. 2022b), we adopt the B-m Inc-n protocol to simulate class-incremental learning. Here, m denotes the number of classes introduced in the initial base session, and n represents the number of new classes added at each subsequent incremental stage. To ensure reproducibility and consistency, we randomly shuffle the class order using a fixed seed (1993), as in (Rebuffi et al. 2017), and apply the same ordering across all methods.

Comparison methods: We compare our approach with several state-of-the-art class-incremental learning (CIL) methods that leverage pre-trained models, including L2P (Wang et al. 2022b), DualPrompt (Wang et al. 2022a), CODA-Prompt (Smith et al. 2023), and SimpleCIL (Zhou et al. 2025a). In addition, we evaluate against recent CLIP-based CIL approaches such as CoOp (Zhou et al. 2022), PROOF (Zhou et al. 2025c), and RAPF (Huang et al. 2024). A naive baseline, denoted as Finetune, directly fine-tunes CLIP on the incremental tasks. All methods are initialized from the same CLIP model to ensure a fair comparison.

Training details: All experiments are conducted using PyTorch (Paszke et al. 2019) on an NVIDIA RTX 4090 GPU. Following prior work (Zhou et al. 2025c; Huang et al. 2024), we adopt the CLIP ViT-B/16 model as the visual backbone across all methods. For vision-only methods that cannot utilize text prompts (e.g., L2P, DualPrompt, CODA-Prompt), we initialize them using CLIP's visual encoder. Unless otherwise stated, we report results using CLIP pre-trained on LAION-400M (Ilharco et al. 2021), with additional results using OpenAI CLIP (Radford et al. 2021) provided in the supplementary material. The learning rate starts from 0.05 and is decayed via cosine annealing. More experimental details are provided in the extended version.

Evaluation metric: Following established protocols (Rebuffi et al. 2017; Zhou et al. 2025c), we evaluate the top-1 accuracy after each incremental stage b, denoted as $\mathcal{A}_b$. We report the final accuracy after the last stage, $\mathcal{A}_B$, as well as the average accuracy across all stages, $\bar{\mathcal{A}} = \frac{1}{B} \sum b = 1^B \mathcal{A}_b$.

Benchmark Comparison

We first evaluate BOFA against a range of state-of-the-art methods on standard class-incremental learning benchmarks. Results are presented in Table 1 and visualized in Figure 2. BOFA consistently outperforms existing approaches across all datasets, demonstrating its robust generalization and continual learning capabilities.

Method	Aircraft				CIFAR100				Cars			
	B0 Inc10		B50 Inc10		B0 Inc10		B50 Inc10		B0 Inc10		B50 Inc10	
	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$
Finetune	3.16	0.96	1.72	1.05	7.84	4.44	5.30	2.46	3.14	1.10	1.54	1.13
CoOp (Zhou et al. 2022)	14.54	7.14	13.05	7.77	47.00	24.24	41.23	24.12	36.46	21.65	37.40	20.87
SimpleCIL (Zhou et al. 2025a)	59.24	48.09	53.05	48.09	84.15	76.63	80.20	76.63	92.04	86.85	88.96	86.85
ZS-CLIP (Radford et al. 2021)	26.66	17.22	21.70	17.22	81.81	71.38	76.49	71.38	82.60	76.37	78.32	76.37
L2P (Wang et al. 2022b)	47.19	28.29	44.07	32.13	82.74	73.03	81.14	73.61	76.63	61.82	76.37	65.64
DualPrompt (Wang et al. 2022a)	44.30	25.83	46.07	33.57	81.63	72.44	80.12	72.57	76.26	62.94	76.88	67.55
CODA-Prompt (Smith et al. 2023)	45.98	27.69	45.14	32.28	82.43	73.43	78.69	71.58	80.21	66.47	75.06	64.19
RAPF (Huang et al. 2024)	50.38	23.61	40.47	25.44	86.14	78.04	82.17	77.93	82.89	62.85	75.87	63.19
BOFA	**69.94**	**59.67**	**65.34**	**60.47**	**86.50**	**79.34**	**83.44**	**79.62**	**93.77**	**89.23**	**91.60**	**89.69**

Method	ImageNet-R				CUB				UCF			
	B0 Inc20		B100 Inc20		B0 Inc20		B100 Inc20		B0 Inc10		B50 Inc10	
	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$
Finetune	1.37	0.43	1.01	0.88	2.06	0.64	0.56	0.47	4.51	1.59	1.21	0.80
CoOp (Zhou et al. 2022)	60.73	37.52	54.20	39.77	27.61	8.57	24.03	10.14	47.85	33.46	42.02	24.74
SimpleCIL (Zhou et al. 2025a)	81.06	74.48	76.84	74.48	83.81	77.52	79.75	77.52	90.44	85.68	88.12	85.68
ZS-CLIP (Radford et al. 2021)	83.37	77.17	79.57	77.17	74.38	63.06	67.96	63.06	75.50	67.64	71.44	67.64
L2P (Wang et al. 2022b)	75.97	66.52	72.82	66.77	70.87	57.93	75.64	66.12	86.34	76.43	83.95	76.62
DualPrompt (Wang et al. 2022a)	76.21	66.65	73.22	67.58	69.89	57.46	74.40	64.84	85.21	75.82	84.31	76.35
CODA-Prompt (Smith et al. 2023)	77.69	68.95	73.71	68.05	73.12	62.98	73.95	62.21	87.76	80.14	83.04	75.03
RAPF (Huang et al. 2024)	81.26	70.48	76.10	70.23	79.09	62.77	72.82	62.93	92.28	80.33	90.31	81.55
BOFA	**85.42**	**79.62**	**82.13**	**79.68**	**86.09**	**79.10**	**82.28**	**79.15**	**93.22**	**88.08**	**92.01**	**88.23**

Method	SUN				Food				ObjectNet			
	B0 Inc30		B150 Inc30		B0 Inc10		B50 Inc10		B0 Inc20		B100 Inc20	
	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$
Finetune	4.51	1.59	0.78	0.72	3.49	1.71	2.14	1.52	1.34	0.47	0.69	0.54
CoOp (Zhou et al. 2022)	45.93	23.11	39.33	24.89	36.01	14.18	33.13	18.67	21.24	6.29	16.21	6.82
SimpleCIL (Zhou et al. 2025a)	82.13	75.58	78.62	75.58	87.89	81.65	84.73	81.65	52.06	40.13	45.11	40.13
ZS-CLIP (Radford et al. 2021)	79.42	72.11	74.95	72.11	87.86	81.92	84.75	81.92	38.43	26.43	31.12	26.43
L2P (Wang et al. 2022b)	82.82	74.54	79.57	73.10	85.66	77.33	80.42	73.13	51.40	39.39	48.91	42.83
DualPrompt (Wang et al. 2022a)	82.46	74.40	79.37	73.02	84.92	77.29	80.00	72.75	52.62	40.72	49.08	42.92
CODA-Prompt (Smith et al. 2023)	83.34	75.71	80.38	74.17	86.18	78.78	80.98	74.13	46.49	34.13	40.57	34.13
RAPF (Huang et al. 2024)	82.13	72.47	78.04	73.10	88.57	81.15	85.53	81.17	48.67	27.43	39.28	28.73
BOFA	**85.62**	**78.87**	**82.58**	**79.49**	**89.01**	**82.74**	**86.06**	**82.95**	**58.04**	**45.14**	**51.07**	**45.37**

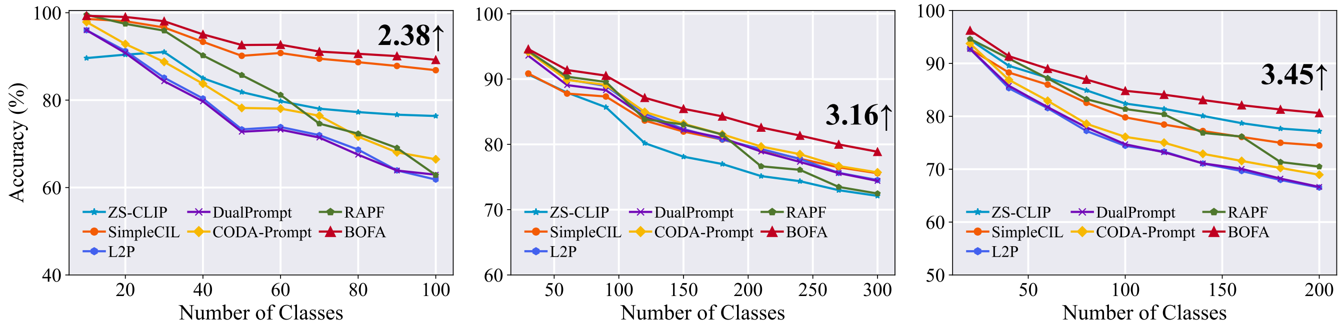
Table 1: Average and last performance comparison of different methods. The best performance is shown in bold. **All methods are initialized with the same pre-trained CLIP without exemplars for a fair comparison.**

Method	Aircraft B0 Inc10		CIFAR100 B0 Inc10		ImageNet-R B0 Inc20		SUN B0 Inc30	
	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$	$\bar{\mathcal{A}}$	$\mathcal{A}_B$
iCaRL	65.04	54.04	77.05	63.7	71.69	58.68	51.44	66.17
MEMO	63.56	52.96	79.22	68.76	60.90	71.78	52.69	66.00
PROOF	61.00	53.59	84.60	76.76	83.03	77.45	81.90	74.66
BOFA	**69.94**	**59.67**	**86.50**	**79.34**	**85.96**	**80.82**	**85.61**	**78.87**

Table 2: Comparison to traditional exemplar-based CIL methods. BOFA does not use any exemplars, whereas the comparative methods retain 10 samples per class. All methods are based on the same pre-trained CLIP.

Among all methods, the naive finetuning baseline performs the worst, suggesting that without proper regularization, the model completely forgets previously learned class representations. Visual prompt-based methods such as L2P, DualPrompt, and CODA-Prompt show limited performance, likely due to their inability to incorporate semantic information from the textual modality. In contrast, CoOp, a textual prompt tuning approach, suffers from substantial degradation in performance, which we attribute to severe forgetting of learned prompts over time. Even compared to recent CLIP-based approaches like RAPF, BOFA achieves significant gains, demonstrating superior robustness against forgetting while preserving the benefits of cross-modal representations.

In addition to exemplar-free methods, we also compare against representative exemplar-based CIL approaches, include iCaRL (Rebuffi et al. 2017), MEMO (Zhou et al. 2023) and PROOF (Zhou et al. 2025c). Table 2 summarizes these results, where all models are initialized with the same CLIP backbone. Notably, despite not using any exemplars, BOFA still surpasses these methods, underscoring the effectiveness of our exemplar-free design.



(a) Cars B0 Inc10 (b) Sun B0 Inc30 (c) ImageNet-R B0 Inc20

Figure 2: Incremental performance of different methods. Accuracy is reported at each incremental stage. BOFA consistently outperforms all baselines, with the final gap to the strongest competitor noted at the end of each curve. Additional results are available in the supplementary material.

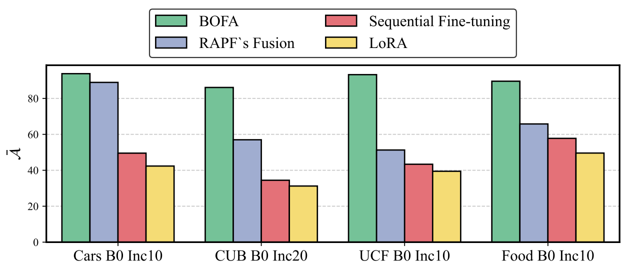
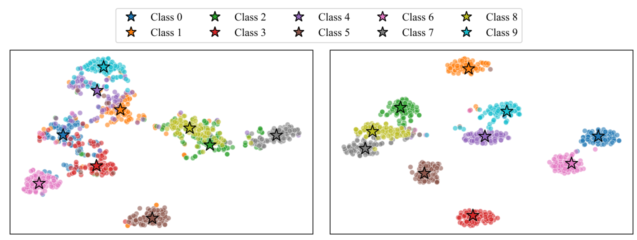


Figure 3: $\bar{A}$ comparison on four datasets for various ablation variants of BOFA .



(a) W/O Orthogonal Low-Rank Fusion (b) W/ Orthogonal Low-Rank Fusion

Figure 4: T-SNE visualization of features (circles) and class prototypes (stars) on CIFAR100 B0 Inc5. We show the feature distributions of old classes (0–4) and new classes (5–9) with (left) and with out (right) applying Orthogonal Low-Rank Fusion.

Further Analysis

Ablation Study of Orthogonal Low-Rank Fusion: To validate our orthogonal low-rank fusion, we benchmarked several methods applied specifically to the cross-modal bridge-layer: sequential fine-tuning (FT), standard LoRA, and an adapted version of RAPF. For RAPF, a state-of-the-art fusion method, we applied its core logic and removed its covariance-based sampling to ensure a fair comparison. As shown in Figure 3, BOFA significantly outperforms all baselines. Notably, our results show FT surpasses standard LoRA, indicating that simple low-rank constraints are not only insufficient for new task adaptation but also fail to effectively mitigate catastrophic forgetting in this context. BOFA is engineered to overcome these dual limitations. It harnesses the adaptive power of FT for a strong initialization, and then rigorously prevents forgetting—a weakness of both FT and LoRA—by constraining this update within the orthogonal safe subspace. This synergy of high plasticity and principled stability enables BOFA to achieve the best overall performance.

Visualizations: We employ t-SNE (Van der Maaten and Hinton 2008) to visualize cross-modal features learned by BOFA on CIFAR100 B0 Inc5 in Figure 4. We compare feature distributions with and without orthogonal low-rank fusion on the second task, plotting features (●) from 5 old and 5 new test classes, along with their corresponding prototypes (★). Without fusion, new class features are well-clustered, but old class features remain entangled and difficult to distinguish. After fusion, features from both old and new classes show clearer separation and better alignment with their prototypes, indicating improved discriminability and prototype consistency.

Conclusion

In this work, we presented BOFA, an effective framework for exemplar-free CIL with CLIP. BOFA fine-tunes only CLIP's cross-modal bridge-layer, using a novel Orthogonal Low-Rank Fusion strategy to mitigate catastrophic forgetting without extra parameters. Furthermore, it employs Cross-Modal Hybrid Prototypes to enhance classification by robustly integrating visual and textual cues. Our results show that this integrated approach effectively preserves knowledge while achieving SOTA discriminative performance.

Acknowledgements

This work is partially supported by NSFC (62506160), NSF of Jiangsu Province (BK 20251251), Fundamental Research Funds for the Central Universities (14380021), JSTJ-2025-147, Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

Aggarwal, C. C. 2018. A survey of stream clustering algorithms. In *Data Clustering*, 231–258. Chapman and Hall/CRC.

Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; and Tuytelaars, T. 2018. Memory aware synapses: Learning what (not) to forget. In *ECCV*, 139–154.

Aljundi, R.; Kelchtermans, K.; and Tuytelaars, T. 2019. Task-free continual learning. In *CVPR*, 11254–11263.

Barbu, A.; Mayo, D.; Alverio, J.; Luo, W.; Wang, C.; Gutfreund, D.; Tenenbaum, J.; and Katz, B. 2019. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. *NeurIPS*, 32.

Bossard, L.; Guillaumin, M.; and Van Gool, L. 2014. Food-101–mining discriminative components with random forests. In *ECCV*, 446–461. Springer.

Chaudhry, A.; Dokania, P. K.; Ajanthan, T.; and Torr, P. H. 2018. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, 532–547.

Chen, S.; Ge, C.; Tong, Z.; Wang, J.; Song, Y.; Wang, J.; and Luo, P. 2022. Adaptformer: Adapting vision transformers for scalable visual recognition. *NeurIPS*, 35: 16664–16678.

De Lange, M.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; and Tuytelaars, T. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE transactions on pattern analysis and machine intelligence*, 44(7): 3366–3385.

Dong, S.; Hong, X.; Tao, X.; Chang, X.; Wei, X.; and Gong, Y. 2021. Few-Shot Class-Incremental Learning via Relation Knowledge Distillation. In *AAAI*, 1255–1263.

Douillard, A.; Cord, M.; Ollion, C.; Robert, T.; and Valle, E. 2020. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, 86–102.

Douillard, A.; Ramé, A.; Couairon, G.; and Cord, M. 2022. Dytox: Transformers for continual learning with dynamic token expansion. In *CVPR*, 9285–9295.

French, R. M. 1999. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4): 128–135.

French, R. M.; and Ferrara, A. 1999. Modeling time perception in rats: Evidence for catastrophic interference in animal learning. In *Proceedings of the 21st Annual Conference of the Cognitive Science Conference*, 173–178. Citeseer.

Gao, Q.; Zhao, C.; Ghanem, B.; and Zhang, J. 2022. R-DFCIL: Relation-Guided Representation Learning for Data-Free Class Incremental Learning. In *ECCV*, 423–439.

Hendrycks, D.; Basart, S.; Mu, N.; Kadavath, S.; Wang, F.; Dorundo, E.; Desai, R.; Zhu, T.; Parajuli, S.; Guo, M.; et al. 2021. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 8340–8349.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hu, E. J.; yelong shen; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Huang, L.; Cao, X.; Lu, H.; and Liu, X. 2024. Class-incremental learning with clip: Adaptive representation adjustment and parameter fusion. In *ECCV*.

Ilharco, G.; Wortsman, M.; Wightman, R.; Gordon, C.; Carlini, N.; Taori, R.; Dave, A.; Shankar, V.; Namkoong, H.; Miller, J.; Hajishirzi, H.; Farhadi, A.; and Schmidt, L. 2021. OpenCLIP.

Jia, C.; Yang, Y.; Xia, Y.; Chen, Y.-T.; Parekh, Z.; Pham, H.; Le, Q.; Sung, Y.-H.; Li, Z.; and Duerig, T. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 4904–4916.

Jia, M.; Tang, L.; Chen, B.; Cardie, C.; Belongie, S. J.; Hariharan, B.; and Lim, S. 2022. Visual Prompt Tuning. In *ECCV*, 709–727.

Jung, D.; Han, D.; Bang, J.; and Song, H. 2023. Generating instance-level prompts for rehearsal-free continual learning. In *ICCV*, 11847–11857.

Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *ICCV Workshop*, 554–561.

Krizhevsky, A. 2009. Learning Multiple Layers of Features from Tiny Images.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 19730–19742.

Li, L.; Peng, J.; Chen, H.; Gao, C.; and Yang, X. 2024. How to configure good in-context sequence for visual question answering. In *CVPR*, 26710–26720.

Li, Z.; and Hoiem, D. 2016. Learning Without Forgetting. In *ECCV*, 614–629.

Lu, Y.; Wang, M.; and Deng, W. 2022. Augmented Geometric Distillation for Data-Free Incremental Person ReID. In *CVPR*, 7329–7338.

Luo, Z.; Liu, Y.; Schiele, B.; and Sun, Q. 2023. Class-incremental exemplar compression for class-incremental learning. In *CVPR*, 11371–11380.

Maji, S.; Rahtu, E.; Kannala, J.; Blaschko, M.; and Vedaldi, A. 2013. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*.

Masana, M.; Liu, X.; Twardowski, B.; Menta, M.; Bagdanov, A. D.; and Van De Weijer, J. 2022. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5): 5513–5533.

Masoudnia, S.; and Ebrahimpour, R. 2014. Mixture of experts: a literature survey. *Artificial Intelligence Review*, 42: 275–293.

McDonnell, M. D.; Gong, D.; Parveneh, A.; Abbasnejad, E.; and Hengel, A. v. d. 2023. RanPAC: Random Projections and Pre-trained Models for Continual Learning. In *NeurIPS*.

Park, J.; Kang, M.; and Han, B. 2021. Class-incremental learning for action recognition in videos. In *ICCV*, 13698–13707.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 8026–8037.

Qi, Z.-H.; Zhou, D.-W.; Yao, Y.; Ye, H.-J.; and Zhan, D.-C. 2025. Adaptive adapter routing for long-tailed class-incremental learning. *Machine Learning*, 114(3): 1–20.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*, 8748–8763.

Rebuffi, S.-A.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. icarl: Incremental classifier and representation learning. In *CVPR*, 2001–2010.

Shi, Y.; Zhou, K.; Liang, J.; Jiang, Z.; Feng, J.; Torr, P. H.; Bai, S.; and Tan, V. Y. 2022. Mimicking the Oracle: An Initial Phase Decorrelation Approach for Class Incremental Learning. In *CVPR*, 16722–16731.

Smith, J. S.; Karlinsky, L.; Gutta, V.; Cascante-Bonilla, P.; Kim, D.; Arbelle, A.; Panda, R.; Feris, R.; and Kira, Z. 2023. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *CVPR*, 11909–11919.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *NIPS*, 4080–4090.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Sun, H.-L.; Zhou, D.-W.; Zhao, H.; Gan, L.; Zhan, D.-C.; and Ye, H.-J. 2025. MOS: Model Surgery for Pre-Trained Model-Based Class-Incremental Learning. In *AAAI*.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *JMLR*, 9(11).

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

Wang, F.-Y.; Zhou, D.-W.; Liu, L.; Ye, H.-J.; Bian, Y.; Zhan, D.-C.; and Zhao, P. 2023a. BEEF: Bi-Compatible Class-Incremental Learning via Energy-Based Expansion and Fusion. In *ICLR*.

Wang, R.; Duan, X.; Kang, G.; Liu, J.; Lin, S.; Xu, S.; Lü, J.; and Zhang, B. 2023b. Attriclip: A non-incremental learner for incremental knowledge learning. In *CVPR*, 3654–3663.

Wang, Y.; Huang, Z.; and Hong, X. 2022. S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. *NeurIPS*, 5682–5695.

Wang, Z.; Zhang, Z.; Ebrahimi, S.; Sun, R.; Zhang, H.; Lee, C.-Y.; Ren, X.; Su, G.; Perot, V.; Dy, J.; et al. 2022a. Dual-prompt: Complementary prompting for rehearsal-free continual learning. In *ECCV*, 631–648.

Wang, Z.; Zhang, Z.; Lee, C.-Y.; Zhang, H.; Sun, R.; Ren, X.; Su, G.; Perot, V.; Dy, J.; and Pfister, T. 2022b. Learning to prompt for continual learning. In *CVPR*, 139–149.

Xiao, J.; Hays, J.; Ehinger, K. A.; Oliva, A.; and Torralba, A. 2010. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 3485–3492. IEEE.

Xu, J.; and Zhu, Z. 2018. Reinforced continual learning. In *NeurIPS*, 899–908.

Yang, X.; Li, Z.; Xu, H.; Zhang, H.; Ye, Q.; Li, C.; Yan, M.; Zhang, Y.; Huang, F.; and Huang, S. 2023. Learning trajectory-word alignments for video-language tasks. In *ICCV*, 2504–2514.

Yoon, J.; Yang, E.; Lee, J.; and Hwang, S. J. 2018. Lifelong Learning with Dynamically Expandable Networks. In *ICLR*.

Yu, J.; Wang, Z.; Vasudevan, V.; Yeung, L.; Seyedhosseini, M.; and Wu, Y. 2024a. Coca: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*.

Yu, J.; Zhuge, Y.; Zhang, L.; Hu, P.; Wang, D.; Lu, H.; and He, Y. 2024b. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *CVPR*, 23219–23230.

Yuan, L.; Chen, D.; Chen, Y.-L.; Codella, N.; Dai, X.; Gao, J.; Hu, H.; Huang, X.; Li, B.; Li, C.; et al. 2021. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*.

Zhao, B.; Xiao, X.; Gan, G.; Zhang, B.; and Xia, S.-T. 2020. Maintaining Discrimination and Fairness in Class Incremental Learning. In *CVPR*, 13208–13217.

Zheng, B.; Zhou, D.-W.; Ye, H.-J.; and Zhan, D.-C. 2025. Task-Agnostic Guided Feature Expansion for Class-Incremental Learning. In *CVPR*.

Zhou, D.-W.; Cai, Z.-W.; Ye, H.-J.; Zhan, D.-C.; and Liu, Z. 2025a. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *International Journal of Computer Vision*, 133: 1012–1032.

Zhou, D.-W.; Cai, Z.-W.; Ye, H.-J.; Zhang, L.; and Zhan, D.-C. 2025b. Dual Consolidation for Pre-Trained Model-Based Domain-Incremental Learning. In *CVPR*.

Zhou, D.-W.; Wang, Q.-W.; Ye, H.-J.; and Zhan, D.-C. 2023. A Model or 603 Exemplars: Towards Memory-Efficient Class-Incremental Learning. In *ICLR*.

Zhou, D.-W.; Zhang, Y.; Wang, Y.; Ning, J.; Ye, H.-J.; Zhan, D.-C.; and Liu, Z. 2025c. Learning without Forgetting for Vision-Language Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Zhou, K.; Yang, J.; Loy, C. C.; and Liu, Z. 2022. Learning to prompt for vision-language models. *IJCV*, 130(9): 2337–2348.

Zhu, F.; Zhang, X.-Y.; Wang, C.; Yin, F.; and Liu, C.-L. 2021. Prototype Augmentation and Self-Supervision for Incremental Learning. In *CVPR*, 5871–5880.