COLA: Continual Orthogonal Low-Rank Adaptation for Class-Incremental Learning

Monu Nagar and Debasis Das

Indian Institute of Technology Jodhpur, India
{nagar.3, debasis}@iitj.ac.in

Abstract. Recent advances in Continual Learning (CL) have adopted foundation models with Low-Rank Adaptation (LoRA) or prompt tuning, utilizing pre-trained representations to achieve more efficient and adaptable learning. However, these methods suffer from two key challenges. First, they learn task-specific adapters, resulting in unbounded parameter expansion as task sequences grow. Second, they maintain replay buffers or store feature representations from previous tasks, which introduces substantial memory overhead and potential privacy risks. To address these limitations, we propose **C**ontinual **O**rthogonal **L**ow-**R**ank **A**daptation (COLA), a novel rehearsal-free, parameter-efficient framework for CL. COLA integrates LoRA's low-rank adaptation with an Oja-inspired learning rule that incrementally approximates the dominant eigenstructure of the feature covariance across tasks. This mechanism continuously tracks the principal directions of prior task knowledge and progressively reduces representational interference between new and previously learned task subspaces. By dynamically adapting a shared low-rank subspace using covariance updates, COLA achieves stable feature projection and continual knowledge integration without additional parameter expansion or memory replay. Extensive experiments on class-incremental benchmarks such as ImageNet-R, ImageNet-A, and CUB200 demonstrate that COLA effectively mitigates catastrophic forgetting while maintaining minimal memory overhead and strong generalization, outperforming existing LoRA-based and prompt-based methods. The code is available at: https://github.com/autovisionproject/COLA.

Keywords: Continual Learning · Low-Rank Adaptation · Forgetting

1 Introduction

Over the past decade, Deep Learning (DL) methods have achieved remarkable success, replicating or surpassing human-level capabilities in various fields [14,43]. However, the dynamic, non-stationary nature of real-world data poses a significant challenge, motivating the development of Class-Incremental Learning (CIL) systems that can continuously adapt to the emergence of new classes [35]. A fundamental challenge in CIL is catastrophic forgetting [24], where training the model on newly introduced classes causes it to lose previously acquired knowledge, leading to significant degradation in performance on earlier classes. This

phenomenon is also known as the stability-plasticity dilemma [16]. Consequently, developing effective strategies to mitigate catastrophic forgetting has become the central focus in designing robust CIL models. The catastrophic forgetting problem in CIL has driven researchers toward pre-trained models (PTMs) as a potential solution [37, 43], based on the assumption that representations learned from large-scale pre-training possess sufficient generalization capacity to resist distributional shifts across incremental tasks. This assumption finds strong empirical support as PTMs trained on massive datasets with substantial computational resources encode diverse features that demonstrably improve CIL performance relative to random initialization. The success of PTM-based approaches has established a new research direction centered on efficient adaptation strategies that utilize pre-trained knowledge [41, 42].

Recent approaches such as L2P [37] and DualPrompt [36] maintain dynamically expanding prompt pools and retrieve task-specific prompts through similarity-based matching with test inputs. CODA-Prompt [28] and Contrastive Prototypical Prompt [17] further enhance prompt selection through joint optimization with CL objectives. While eliminating the requirement for explicit task labels, these techniques incur inference overhead that scales with the number of tasks. To address this limitation, HiDe-Prompt [33] adopts memory replay strategies, storing substantial numbers of exemplars to support incremental prompt learning. Similarly, InfLoRA [18] employs Low-Rank Adaptation (LoRA) [10] for parameter efficiency while still requiring extensive sample storage during sequential LoRA updates. SD-LoRA [38] attempts to mitigate the need for replay by decoupling magnitude and directional learning within LoRA components; however, it accumulates task-specific LoRA parameters across the learning sequence to inform subsequent task adaptation. These memory-intensive strategies compromise scalability in practical applications, especially in resource-limited environments or in extended CL scenarios. Table 1 characterizes four critical properties that define an effective CL approach integrating foundation models:

1. **Rehearsal-free.** The approach should operate without storing exemplars from prior tasks, so that learning remains scalable as task sequences grow.
2. **Inference Efficiency.** The method should maintain a minimal inference cost regardless of the number of tasks encountered, ensuring long-term computational efficiency and scalable deployment.
3. **End-to-End Optimization.** All learnable components should be jointly optimized toward CL objectives through unified training, rather than through decoupled or multi-stage optimization procedures.
4. **Memory Efficiency.** The approach should keep parameter growth and storage overhead minimal as the number of tasks increases.

To achieve these properties, we propose the COLA framework, which integrates low-rank adaptation with covariance-guided subspace orthogonalization. Specifically, COLA incrementally estimates the dominant eigenstructure of feature activations by updating task-wise covariance matrices (Σ_t) and consolidating them into a global covariance (Σ_{global}), thereby preserving prior knowledge within

Table 1: Comparison of CIL methods across four key properties: rehearsal-free learning, inference efficiency, end-to-end optimization, and memory efficiency.

Models	Year	Rehearsal-free	Runtime Efficiency	End-to-end Optimization	Memory Efficiency
L2P [37]	(CVPR, 2022)	✓	×	×	✓
DualPrompt [36]	(ECCV, 2022)	✓	×	×	×
CODA-Prompt [28]	(CVPR, 2023)	✓	✓	×	✓
RanPAC [22]	(NeurIPS, 2023)	✓	✓	×	×
HiDe-Prompt [33]	(WACV, 2024)	×	×	✓	×
InfLoRA [18]	(CVPR, 2024)	×	✓	✓	×
PLAN [34]	(ICCV, 2025)	✓	✓	✓	×
CoSO [3]	(NeurIPS, 2025)	✓	✓	✓	×
SD-LoRA [38]	(ICLR, 2025)	✓	✓	✓	×
Ours (COLA)		✓	✓	✓	✓

the model's representational statistics. These covariance-guided updates dynamically project new gradient directions onto orthogonal subspaces within the shared LoRA rank, preventing interference with previously learned tasks. By maintaining knowledge in the evolving eigenbasis rather than through exemplar replay, COLA achieves rehearsal-free continual learning. During inference, the model employs consolidated low-rank adapters without task-identity-based routing, preserving computational efficiency. All LoRA parameters and covariance statistics are jointly optimized within a unified learning objective, achieving true end-to-end training. Furthermore, COLA maintains constant memory overhead by sharing a fixed-size LoRA adapter across all tasks, regardless of the number of tasks encountered. Unlike OGD [8], GPM [26], and InfLoRA [18], which store per task subspaces and require SVD at each task boundary, COLA maintains a single fixed size covariance summary updated online, replacing hard orthogonality with a soft projection mechanism. Our main contributions are summarized as follows:

1. We propose Continual Orthogonal Low-Rank Adaptation (COLA), a novel rehearsal-free continual learning framework that integrates low-rank adaptation with covariance-guided orthogonal projections, ensuring interference-free updates and robust knowledge retention.
2. The proposed method incrementally aligns task-specific representations with global covariance statistics, resulting in soft orthogonal subspace constraints that preserve prior knowledge and effectively mitigate forgetting.
3. We conduct extensive experiments across multiple CL benchmarks and diverse datasets, demonstrating the effectiveness and scalability of COLA. Empirically, COLA achieves a 4.8% ($\mathcal{AAA}$) improvement on ImageNet-R (25 tasks), maintaining a fixed parameter budget with no inference overhead.

2 Related Work

In this section, we discuss prior methods most closely related to our work, focusing on continual learning strategies and parameter-efficient adaptation mechanisms.

2.1 Continual Learning (CL)

CL approaches can be organized into three main categories: rehearsal-based, regularization-based, and architecture-based methods. Rehearsal-based approaches

[19, 21, 25, 40] store representative samples from previous tasks and leverage knowledge distillation to preserve historical knowledge in the current model. Regularization-based methods [27, 29, 39] prevent forgetting by imposing constraints on parameter updates, thereby stabilizing learned features across tasks. Architecture-based techniques [11, 15, 31, 36] adapt to new tasks through dynamic network expansion or task-specific parameter allocation, enabling flexible structural evolution. Despite their efficacy in mitigating forgetting, rehearsal-based approaches require substantial storage capacity and pose privacy risks. Regularization-based methods preserve parameter stability but constrain model plasticity, hindering novel task integration. Architecture-based solutions that augment network capacity face scalability concerns due to continuous structural expansion. Recent work addresses these limitations through parameter-efficient fine-tuning, which updates only a small number of parameters while leveraging pre-trained model knowledge.

2.2 Parameter-Efficient Fine-Tuning (PEFT) in CL

Foundation models have emerged as a powerful approach for continual learning, effectively mitigating catastrophic forgetting through their inherent ability to generalize and transfer knowledge across tasks [18, 28, 33]. The integration of PEFT techniques with foundation models has become essential, as full fine-tuning of large-scale architectures for each incremental task remains computationally expensive [6]. PEFT techniques mainly include low-rank adapters that insert compact learnable modules into Transformer layers [13, 32], prompt-tuning [4], and prefix-tuning [1] that introduce learnable input representations and Low-Rank Adaptation (LoRA) [10], which augments pre-trained weights with low-rank parameter updates. Additionally, Vision Transformers (ViTs) with LoRA have demonstrated strong potential for CL, attributed to their superior generalization ability and highly transferable feature representations. L2P [37], DualPrompt [36], and CODA-Prompt [28] combine ViTs with prompt-tuning strategies to mitigate forgetting as new tasks arrive. Building upon this foundation, HiDe-Prompt [33] and InfLoRA [18] achieve enhanced performance through extensive sample rehearsal, though at the cost of increased memory overhead. Furthermore, projection-based methods such as Orthogonal Gradient Descent [8] and Gradient Projection Memory [26] explicitly construct orthogonal gradient subspaces and enforce strict projection of new updates onto these subspaces; however, these methods rely on accurate subspace estimation, making them sensitive to noise and distribution shifts as stored bases accumulate over tasks.

More recent methods, such as CoSO [3] and VPT-NSP2 [20], also depend on explicit subspace decomposition and task-specific parameter expansions, which introduce instability as task complexity increases. VPT-NSP2 introduces a null-space approximation to achieve prompt-gradient orthogonal projection, mitigating feature distribution shift caused by LayerNorm in the transformer block, whereas CoSO applies SVD to mitigate catastrophic forgetting by projecting gradient updates onto sequential subspaces orthogonal to those of previous tasks. In

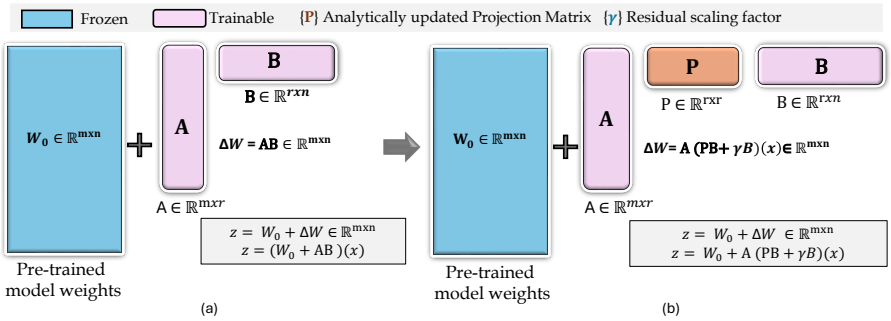

Fig. 1: Illustration of parameter update mechanisms for continual learning. (a) Vanilla LoRA performs direct low-rank adaptation without task separation. (b) Proposed COLA integrates orthogonal projection operators derived from global and task-specific covariance matrices with LoRA to align new task features in orthogonal subspaces.

contrast, COLA does not enforce hard orthogonality constraints or store task-specific subspaces. Instead, it employs an Oja-inspired online update rule to incrementally estimate principal adaptation directions, thereby ensuring stable convergence in streaming covariance estimation. The covariance statistics act as a soft guidance signal for adapting LoRA parameters rather than a strict projection constraint, meaning that minor estimation bias does not directly restrict gradient updates, improving robustness to evolving data distributions and class imbalance while maintaining computational efficiency across longer task sequences. Despite recent progress, existing foundation model-based CL methods fail to satisfy all desirable properties highlighted in Table 1 simultaneously. SD-LoRA [38] overcomes rehearsal dependency through model-merging techniques; however, it lacks task-similarity awareness and suffers from task-proportional memory growth due to accumulated task-specific adapters. These limitations motivate COLA, which addresses all desired properties through fixed-capacity adaptation with covariance-guided orthogonalization.

3 Proposed Method

This section introduces the COLA method, starting with the problem formulation and followed by the core idea.

3.1 Problem Definition

We consider the rehearsal-free Class-Incremental Learning (CIL) setting, where data arrive in a sequence of T tasks, denoted as $\mathcal{D} = \{\mathcal{D}_t\}_{t=1}^{T}$. Each task t consists of a dataset $\mathcal{D}_t = \{\mathcal{X}_t, \mathcal{Y}_t\}$, where $\mathcal{X}_t = \{x_{t,j}\}_{j=1}^{n_t}$ is the input set and $\mathcal{Y}_t = \{y_{t,j} \in \mathcal{C}_t\}_{j=1}^{n_t}$ is the corresponding label set comprising n_t samples. Here, $x_{t,j}$ represents the j^{th} input image and $\mathcal{C}_t$ denotes the class set for task t. Class

sets across tasks are disjoint, i.e., $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$ for $i \neq j$. At each stage t, the model comprises a feature extractor $f_t : \mathbb{R}^{h\times w\times 3} \to \mathbb{R}^d$ and a classification head $g_t : \mathbb{R}^d \to \mathbb{R}^{l_t}$, where d is the feature dimension and $l_t = \sum_{j=1}^{t} |\mathcal{C}_j|$ is the cumulative number of learned classes. Given an input x, the model predicts:

$$\hat{y} = \arg\max(g_t \circ f_t)(x) \tag{1}$$

3.2 Overview of COLA

We introduce Continual Orthogonal Low-Rank Adaptation (COLA), a rehearsal-free framework that enables efficient, scalable, and memory-conscious adaptation of pre-trained vision transformers in the CIL setting. COLA achieves this by introducing three main components: (1) lightweight low-rank adaptation modules, (2) orthogonal subspace alignment through covariance-guided projection, and (3) continual covariance consolidation to preserve past knowledge. An overview of the propose framework is illustrated in Fig. 1.

Low-Rank Adaptation (LoRA). Given a frozen pre-trained ViT backbone with parameters θ_0, we insert shared low-rank adapters into each attention layer and reuse them across tasks, enabling efficient continual learning without full fine-tuning. For layer $i \in \{1, \ldots, L\}$, let $\mathbf{W}_{\text{qkv}}^{(i)} \in \mathbb{R}^{d\times 3d}$ denote the frozen QKV projection. We introduce learnable low-rank matrices:

$$\mathbf{A}_q^{(i,t)}, \mathbf{A}_v^{(i,t)} \in \mathbb{R}^{d\times r} \tag{2}$$

$$\mathbf{B}_q^{(i,t)}, \mathbf{B}_v^{(i,t)} \in \mathbb{R}^{r\times d} \tag{3}$$

where $r \ll d$. For an input $\mathbf{x} \in \mathbb{R}^d$, the low-rank features are:

$$\mathbf{z}_q^{(i,t)} = \mathbf{A}_q^{(i,t)}\mathbf{x}, \quad \mathbf{z}_v^{(i,t)} = \mathbf{A}_v^{(i,t)}\mathbf{x}. \tag{4}$$

Orthogonality-Aware Subspace Alignment. A key challenge in CIL is catastrophic forgetting caused by representational interference between sequentially learned tasks. To mitigate this, COLA introduces an adaptive subspace alignment mechanism guided by covariance statistics that progressively steers new task features toward subspaces complementary to previously learned knowledge, rather than imposing strict orthogonality constraints. For each layer i, we maintain global covariance ($\boldsymbol{\Sigma}_{\text{global}}^{(i)}$) that captures shared statistics across all past tasks, and a task covariance ($\boldsymbol{\Sigma}_{\text{task}}^{(i,t)}$) that captures the feature statistics of the current task, Both are initialized as $\boldsymbol{\Sigma}_{\text{global}}^{(i)} = \boldsymbol{\Sigma}_{\text{task}}^{(i,t)} = \epsilon\mathbf{I}_r$, where $\epsilon > 0$ ensures numerical stability. We then define the projection matrix:

$$\mathbf{P}^{(i,t)} = \mathbf{I}_r - \alpha \frac{\boldsymbol{\Sigma}_{\text{global}}^{(i)}}{\|\boldsymbol{\Sigma}_{\text{global}}^{(i)}\|_F}, \quad \mathbf{P} \in \mathbb{R}^{r\times r} \tag{5}$$

where $\alpha > 0$ is a scaling factor and $\mathbf{I}_r \in \mathbb{R}^{r\times r}$ is the identity matrix. Although $\mathbf{P}$ is not a strict projector, it acts as a soft alignment operator along the dominant

Algorithm 1 COLA Framework

Require: Sequence of tasks $\mathcal{D} = \{\mathcal{D}_t\}_{t=1}^{T}$, ViT backbone f with parameters θ_0, layers L, rank r.
Require: Hyperparameters: η, β, λ, α, K, ϵ.
Initialize: For each layer $i \in \{1, \ldots, L\}$:
  Initialize $\boldsymbol{\Sigma}_{\text{global}}^{(i)}, \boldsymbol{\Sigma}_{\text{task}}^{(i)} \leftarrow \epsilon \mathbf{I}_r$ and γ
for $t = 1$ **to** T **do**
  for each mini-batch $\{\mathbf{x}_j, \mathbf{y}_j\}_{j=1}^{B}$ in $\mathcal{D}_t$ **do**
    for each layer $i = 1$ **to** L **do**
      Compute low-rank features $\mathbf{z}_q^{(i,t)}, \mathbf{z}_v^{(i,t)}$ Eq.(4)
      Construct soft-orthogonal projection $\mathbf{P}^{(i,t)}$, Eq. (5)
      Apply projection to obtain $\tilde{\mathbf{z}}_q^{(i,t)}, \tilde{\mathbf{z}}_v^{(i,t)}$, Eq. (6)
      Compute batch covariance $\mathbf{C}_{\text{batch}}^{(i,t)}$, Eq. (7)
      Update task covariance $\boldsymbol{\Sigma}_{\text{task}}^{(i,t)}$, Eq. (8)
      Generate adapted features $\hat{\mathbf{z}}_q^{(i,t)}, \hat{\mathbf{z}}_v^{(i,t)}$, Eq. (10) - (11)
      Compute final QKV projections, Eq. (12)
    end for
    Compute loss $\mathcal{L}_t$ and update $\{\mathbf{A}^{(i,t)}, \mathbf{B}^{(i,t)}\}_{i=1}^{L}, \gamma$
    if batch count $\mod K = 0$ **then**
      Consolidate global covariance for all layers using Eq. (9)
    end if
  end for
  //**Task Transition.**
  if $t < T$ **then**
    Update: $\boldsymbol{\Sigma}_{\text{global}}^{(i)} \leftarrow \boldsymbol{\Sigma}_{\text{global}}^{(i)} + \boldsymbol{\Sigma}_{\text{task}}^{(i,t)}$ for all i
    Reset task covariance: $\boldsymbol{\Sigma}_{\text{task}}^{(i,t+1)} \leftarrow \epsilon \mathbf{I}_r$ for all i
  end if
end for
Output: Model parameters $\{\mathbf{A}^{(i,T)}, \mathbf{B}^{(i,T)}\}_{i=1}^{L}$ and global covariance $\{\boldsymbol{\Sigma}_{\text{global}}^{(i)}\}_{i=1}^{L}$.

eigenvectors of $\boldsymbol{\Sigma}_{\text{global}}^{(i)}$, ensuring stability under streaming covariance updates. Using the low-rank features from (4), the projected features are defined as:

$$\tilde{\mathbf{z}}_q^{(i,t)} = \mathbf{P}^{(i,t)} \mathbf{z}_q^{(i,t)}, \quad \tilde{\mathbf{z}}_v^{(i,t)} = \mathbf{P}^{(i,t)} \mathbf{z}_v^{(i,t)} \tag{6}$$

Covariance-Guided Knowledge Consolidation. To incrementally update the covariance structures while avoiding catastrophic forgetting, COLA maintains low-dimensional covariance summaries of the adapted feature space and updates them online. These covariance statistics capture the dominant task-specific features, which we then use to construct soft-orthogonal projection operators that reduce interference with previously learned tasks.

Let $\{\tilde{\mathbf{z}}_{q,j}^{(i,t)}\}_{j=1}^{B}$ be the batch of projected low-rank query features at layer i for the current mini-batch. We first compute the batch covariance:

$$\mathbf{C}_{\text{batch}}^{(i,t)} = \frac{1}{B} \sum_{j=1}^{B} \left(\tilde{\mathbf{z}}_{q,j}^{(i,t)} - \bar{\mathbf{z}}_q^{(i,t)}\right) \left(\tilde{\mathbf{z}}_{q,j}^{(i,t)} - \bar{\mathbf{z}}_q^{(i,t)}\right)^{\top} \tag{7}$$

where $\bar{\mathbf{z}}_q^{(i,t)} = \frac{1}{B}\sum_{j=1}^{B} \tilde{\mathbf{z}}_{q,j}^{(i,t)}$ is the batch mean. Eq. (7) extracts the second-order statistics of the current batch, approximating the local task covariance in the low-rank feature space. We then update the *task-specific* covariance using [23]:

$$\boldsymbol{\Sigma}_{\text{task}}^{(i,t)} \leftarrow \boldsymbol{\Sigma}_{\text{task}}^{(i,t)} + \eta\big(\mathbf{C}_{\text{batch}}^{(i,t)} - \beta\boldsymbol{\Sigma}_{\text{task}}^{(i,t)}\big) \tag{8}$$

where $\eta > 0$ is a learning rate and $\beta > 0$ is a small decay factor. Eq. (8) performs a streaming accumulation of principal directions for the current task: repeated application causes $\boldsymbol{\Sigma}_{\text{task}}^{(i,t)}$ to emphasize dominant eigenvectors of the feature covariance (i.e., the task subspace), while the decay term prevents unbounded growth and reduces sensitivity to noisy batches. This update follows Oja's rule [23], ensuring that $\boldsymbol{\Sigma}_{\text{task}}^{(i,t)}$ progressively captures the principal eigenstructure of the task's feature covariance. For every K batches, we consolidate task-level statistics into the *global* covariance as follows:

$$\boldsymbol{\Sigma}_{\text{global}}^{(i)} \leftarrow \boldsymbol{\Sigma}_{\text{global}}^{(i)} + \lambda\big(\boldsymbol{\Sigma}_{\text{task}}^{(i,t)} - \boldsymbol{\Sigma}_{\text{global}}^{(i)}\big) \tag{9}$$

with consolidation rate $\lambda \in (0, 1)$. The resulting global covariance matrix, which captures the structure of previously learned subspaces, enables COLA to construct soft projection operators that redirect new task features into subspaces orthogonal to those of prior tasks, reducing backward interference in a manner analogous to subspace-based continual learning methods that guide updates away from previously occupied feature directions [8, 26]. The adapted and original low-rank features are then combined as:

$$\hat{\mathbf{z}}_q^{(i,t)} = \tilde{\mathbf{z}}_q^{(i,t)} + \gamma\mathbf{z}_q^{(i,t)} \tag{10}$$

$$\hat{\mathbf{z}}_v^{(i,t)} = \tilde{\mathbf{z}}_v^{(i,t)} + \gamma\mathbf{z}_v^{(i,t)} \tag{11}$$

Here, $\tilde{\mathbf{z}}$ guides the representation into directions complementary to prior subspaces, while $\gamma\mathbf{z}$ preserves task-specific information through the residual scaling factor (γ), which continuously balances the stability-plasticity trade-off. The adapted projections are subsequently integrated into the attention qkv computation:

$$\begin{bmatrix} \mathbf{q}^{(i,t)} \\ \mathbf{k}^{(i,t)} \\ \mathbf{v}^{(i,t)} \end{bmatrix} = \mathbf{W}_{\text{qkv}}^{(i)}\mathbf{x} + \begin{bmatrix} \mathbf{B}_q^{(i,t)}\hat{\mathbf{z}}_q^{(i,t)} \\ \mathbf{0} \\ \mathbf{B}_v^{(i,t)}\hat{\mathbf{z}}_v^{(i,t)} \end{bmatrix} \tag{12}$$

At each task t, the model is trained using a cross-entropy loss on current-task samples, with the classification head spanning all previously encountered classes:

$$\mathcal{L}_t = -\frac{1}{n_t}\sum_{j=1}^{n_t} \log p(y_{t,j} \mid x_{t,j}; \theta_0, \{\mathbf{A}^{(i,t)}, \mathbf{B}^{(i,t)}\}_{i=1}^{L}) \tag{13}$$

where only the low-rank parameters $(\mathbf{A}^{(i,t)}, \mathbf{B}^{(i,t)})$ are optimized, while the backbone parameters (θ_0) remain frozen and covariance statistics are updated through gradient-free accumulation. When transitioning to a new task, COLA consolidates the covariance ($\boldsymbol{\Sigma}_{\text{global}}^{(i)} \leftarrow \boldsymbol{\Sigma}_{\text{global}}^{(i)} + \boldsymbol{\Sigma}_{\text{task}}^{(i,t)}$ for all i), resets task statistics ($\boldsymbol{\Sigma}_{\text{task}}^{(i,t+1)} \leftarrow \epsilon\mathbf{I}$ for all i), and continues training the existing low-rank adapters on the next task.

4 Experimental Setup

This section details the benchmark datasets, evaluation protocols, and implementation settings used to validate COLA.

Implementation Details. To ensure rigorous and fair evaluation, we utilize a frozen ViT-B/16 [7] backbone initialized with ImageNet-21K pretraining and ImageNet-1K fine-tuning weights. COLA integrates low-rank adaptation modules into the query and value projections of all attention layers, with the rank fixed to 10. For the covariance-driven projection mechanism, we set the learning rate η to 10^{-4}, the projection decay factor β to 0.01, the consolidation rate λ to 0.3, the consolidation frequency K to 10 batches, and the projection scale factor α to 0.02. Following established protocols [38], we maintain consistent hyperparameter settings across all baseline comparisons. Task-specific LoRA parameters are trained sequentially, with orthogonal projection guidance provided by incrementally updated covariance matrices. All experiments are conducted using PyTorch on an NVIDIA L40S GPU. Training employs the Adam optimizer with a learning rate of 0.008 and a batch size of 128, running for 30 epochs on ImageNet-R and 20 epochs on other datasets. All results are reported as *mean* $\pm$ *standard* deviation (%) over four independent runs conducted with different random seeds to ensure statistical reliability.

Dataset. We conduct experiments on three widely used CL benchmarks: ImageNet-R [2], ImageNet-A [9], and CUB200 [30]. ImageNet-R comprises 200 ImageNet classes [5] depicted in various artistic renderings, providing diversity in visual styles. ImageNet-A contains 200 classes featuring naturally occurring adversarial examples that challenge models trained on standard ImageNet data. CUB200 offers fine-grained classification challenges with 11,788 bird images spanning 200 species. Following standard practice [12, 18], we partition ImageNet-R and ImageNet-A into 10 sequential tasks with 20 classes each, while CUB200 is similarly divided into 10 tasks of 20 species per task. Additionally, we evaluate ImageNet-R under 20-task (10 classes each) and 25-task (8 classes each) configurations to assess scalability over longer sequences.

Evaluation Metrics. To assess the efficacy of our proposed methodology, we adopt three standard and widely used CL evaluation metrics consistent with the evaluation protocol in [18, 37]. ***Average Accuracy*** ($\mathcal{A}_{cc}$), ***Average Anytime Accuracy*** ($\mathcal{AAA}$), and **Forgetting** ($\mathcal{F}$). The $\mathcal{A}_{cc}$ metric quantifies the overall performance after completing all T tasks and is defined as $\mathcal{A}_{cc} = \frac{1}{T}\sum_{i=1}^{T} a_{T,i}$, where $a_{T,i}$ denotes the accuracy on the i-th task after training on the final (T-th) task. The $\mathcal{AAA}$ metric captures the model's performance throughout the learning process by averaging the accuracy over all tasks encountered up to each stage: $\mathcal{AAA} = \frac{1}{T}\sum_{i=1}^{T}\left(\frac{1}{i}\sum_{j=1}^{i} a_{i,j}\right)$. Here, $a_{i,j}$ represents the accuracy on the j-th task after training on the i-th task. Finally, the Forgetting metric quantifies catastrophic forgetting by measuring the average accuracy drop between the best accuracy achieved for each task during training and its final accuracy after all tasks have been learned: $\mathcal{F} = \frac{1}{T-1}\sum_{i=1}^{T-1}(\max_{t\leq T} a_{t,i} - a_{T,i})$.

Table 2: Performance comparison of COLA against state-of-the-art continual learning methods for $\mathcal{A}_{cc}$ and $\mathcal{AAA}$ metrics on ImageNet-R across varying task lengths. All methods use identical hyperparameters and a fixed ViT-B/16 backbone with a LoRA rank of 10 for fair comparison. All results are reported as *mean* ± *standard* deviation.

Method	ImageNet-R (10 Task)		ImageNet-R (20 Task)		ImageNet-R (25 Task)	
	$\mathcal{A}_{cc}$(%)	$\mathcal{AAA}$(%)	$\mathcal{A}_{cc}$(%)	$\mathcal{AAA}$(%)	$\mathcal{A}_{cc}$(%)	$\mathcal{AAA}$(%)
Full Fine-Tuning	$60.57_{\pm 1.3}$	$72.31_{\pm 1.23}$	$49.95_{\pm 1.31}$	$65.32_{\pm 0.92}$	$40.12_{\pm 1.43}$	$46.11_{\pm 0.98}$
L2P [37]	$71.26_{\pm 0.65}$	$76.13_{\pm 0.86}$	$68.97_{\pm 0.57}$	$74.16_{\pm 0.16}$	$60.11_{\pm 1.09}$	$65.12_{\pm 0.94}$
DualPrompt [36]	$68.22_{\pm 0.30}$	$73.81_{\pm 0.39}$	$65.23_{\pm 0.45}$	$71.30_{\pm 0.15}$	$58.00_{\pm 0.66}$	$64.32_{\pm 0.76}$
CODA-Prompt [28]	$74.05_{\pm 0.41}$	$78.14_{\pm 0.39}$	$69.38_{\pm 0.34}$	$73.95_{\pm 0.76}$	$62.92_{\pm 0.34}$	$67.11_{\pm 0.49}$
HiDe-Prompt [33]	$74.65_{\pm 0.14}$	$78.46_{\pm 0.28}$	$73.59_{\pm 0.21}$	$77.93_{\pm 0.38}$	$67.11_{\pm 0.43}$	$70.56_{\pm 0.56}$
RanPAC [22]	$74.58_{\pm 0.77}$	$81.67_{\pm 0.48}$	$72.40_{\pm 0.60}$	$79.27_{\pm 0.40}$	$71.17_{\pm 1.56}$	$74.11_{\pm 0.32}$
InfLoRA [18]	$74.75_{\pm 0.65}$	$80.67_{\pm 0.56}$	$69.89_{\pm 0.66}$	$76.68_{\pm 0.76}$	$68.01_{\pm 0.44}$	$75.12_{\pm 0.16}$
VPT-NSP2 [20]	$78.25_{\pm 0.56}$	$79.45_{\pm 0.76}$	$74.68_{\pm 0.82}$	$79.85_{\pm 0.61}$	$67.12_{\pm 0.34}$	$75.31_{\pm 0.46}$
PLAN [34]	$75.25_{\pm 0.56}$	$80.41_{\pm 0.33}$	$73.25_{\pm 0.42}$	$78.41_{\pm 0.26}$	$71.43_{\pm 0.54}$	$75.23_{\pm 0.45}$
CoSO [3]	$76.52_{\pm 0.87}$	$83.12_{\pm 0.76}$	$72.52_{\pm 0.34}$	$79.67_{\pm 0.65}$	$67.35_{\pm 0.45}$	$77.24_{\pm 0.43}$
SD-LoRA [38]	$77.04_{\pm 0.51}$	$82.55_{\pm 1.50}$	$74.96_{\pm 0.40}$	$80.02_{\pm 0.69}$	$70.78_{\pm 0.43}$	$74.71_{\pm 0.54}$
COLA (Ours)	**77.56** $_{\pm 0.35}$	**83.40**$_{\pm 0.38}$	**76.12**$_{\pm 0.44}$	**82.27**$_{\pm 0.28}$	**76.01**$_{\pm 0.38}$	**82.11**$_{\pm 0.47}$

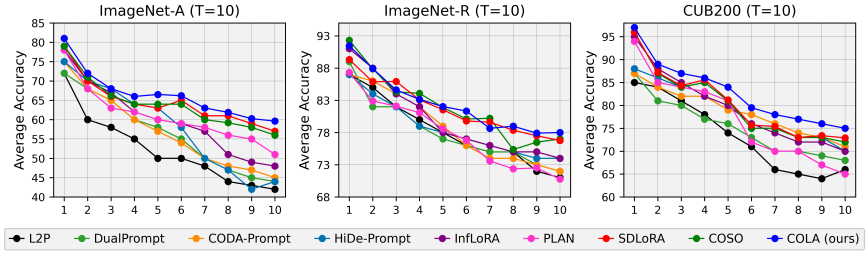

Fig. 2: Comparison of average accuracy (%) across methods during sequential training on (a) ImageNet-A, (b) ImageNet-R, and (c) CUB200 datasets with ViT-B/16 backbone.

5 Model Comparison

We evaluate COLA against existing CL methods built upon Vision Transformer architectures. The baseline methods include prompt-based techniques such as L2P [37], DualPrompt [36], CODA-Prompt [28], and HiDe-Prompt [33], and parameter-efficient low-rank adaptation approaches including InfLoRA [18], PLAN [34], SD-LoRA [38] and CoSO [3]. To maintain experimental consistency, we adopt ViT-B/16 as the shared backbone architecture across all competing methods. Tables 2 and 3 present performance across three benchmark datasets with different characteristics: ImageNet-A with adversarial distribution shifts, ImageNet-R with artistic domain variations, and CUB200 with fine-grained visual discrimination. On ImageNet-A and CUB200, COLA achieves $\mathcal{A}_{\text{cc}}$ of 59.20% and 74.51% respectively after 10 tasks, outperforming SD-LoRA. Furthermore, on ImageNet-R, COLA maintains an average anytime accuracy of 82.11% over a 25-task sequence compared to SD-LoRA's 74.71%, and achieves a 4.87% improvement over CoSO, demonstrating its strong resistance to catas-

Table 3: Performance comparison on ImageNet-A and CUB200 benchmarks. Results (%) are reported as *mean* ± *standard* deviation over four independent runs.

Method	ImageNet-A (10 Task)		CUB200 (10 Task)	
	$\mathcal{A}_{cc}$	$\bar{\mathcal{A}}$	$\mathcal{A}_{cc}$	$\bar{\mathcal{A}}$
L2P [37]	$42.94_{\pm 1.22}$	$51.40_{\pm 1.96}$	$65.18_{\pm 2.46}$	$76.12_{\pm 1.29}$
DualPrompt [36]	$45.49_{\pm 0.90}$	$54.68_{\pm 1.29}$	$68.00_{\pm 1.20}$	$79.40_{\pm 0.96}$
CODA-Prompt [28]	$45.36_{\pm 0.78}$	$57.06_{\pm 0.96}$	$71.92_{\pm 0.45}$	$78.76_{\pm 0.77}$
HiDe-Prompt [33]	$42.70_{\pm 0.67}$	$56.32_{\pm 0.48}$	$69.12_{\pm 1.14}$	$77.46_{\pm 1.01}$
InfLoRA [18]	$49.20_{\pm 1.12}$	$60.92_{\pm 0.76}$	$70.82_{\pm 0.44}$	$81.39_{\pm 0.16}$
PLAN [34]	$54.12_{\pm 0.15}$	$61.45_{\pm 0.76}$	$67.14_{\pm 0.54}$	$76.26_{\pm 1.17}$
CoSO [3]	$56.43_{\pm 0.16}$	$65.54_{\pm 1.20}$	$72.23_{\pm 0.87}$	$81.62_{\pm 1.67}$
SD-LoRA [38]	$57.76_{\pm 0.51}$	$66.56_{\pm 1.50}$	$72.90_{\pm 0.79}$	$82.07_{\pm 1.12}$
COLA (Ours)	$\mathbf{59.20}_{\pm 0.87}$	$\mathbf{67.46}_{\pm 0.58}$	$\mathbf{74.51}_{\pm 0.88}$	$\mathbf{84.38}_{\pm 0.97}$

Table 4: Comparison of computational cost and memory footprint across methods, reporting GFLOPs per forward pass, learnable parameters, and stored features.

Method	GFLOPs ↓	Learnable Params (M) ↓	Stored Features (M) ↓
L2P [37]	70.14	0.48	0.00
DualPrompt [36]	70.26	0.06	0.00
CODA-Prompt [28]	70.61	0.38	0.00
HiDe-Prompt [33]	70.36	0.08	0.15
InfLoRA [18]	35.12	0.37	0.10
CoSO [3]	35.12	7.09	0.00
SD-LoRA [38]	35.12	0.37	0.00
COLA (Ours)	**32.22**	**0.37**	**0.00**

trophic forgetting across longer task sequences. Although CoSO and SD-LoRA remain the strongest baselines, both show fundamental limitations that become increasingly pronounced over longer task sequences. CoSO prevents forgetting by constraining gradient updates to sequentially derived SVD subspaces; however, this approach introduces task interference and performance degradation as the number of tasks grows. SD-LoRA, on the other hand, incrementally adds decoupled low-rank components with separate magnitude and direction learning, resulting in a high computational cost and memory demand that compound with each additional task. In contrast, COLA maintains a compact low-rank adapter that is continuously refined through covariance-guided updates, without expensive decompositions or expanding LoRA components, thereby reducing memory overhead and improving scalability across longer task sequences. In addition to outperforming LoRA-based methods, COLA also surpasses prompt-based methods such as DualPrompt, which achieves 68.22% $\mathcal{A}_{\text{cc}}$ on ImageNet-R compared to COLA's 77.56%, demonstrating that discrete prompt selection fails to capture continuous feature distributions across diverse tasks.

Fig. 2 illustrates the evolution of accuracy across 10 sequential tasks. COLA consistently achieves the highest initial task accuracy across all datasets, attaining 81.21% on ImageNet-A, 91.47% on ImageNet-R, and 97.02% on CUB200, outperforming the closest state-of-the-art methods, SD-LoRA and InfLoRA. More critically, COLA exhibits the most gradual performance degradation as new tasks are introduced. On ImageNet-R, COLA maintains 77.56% accuracy

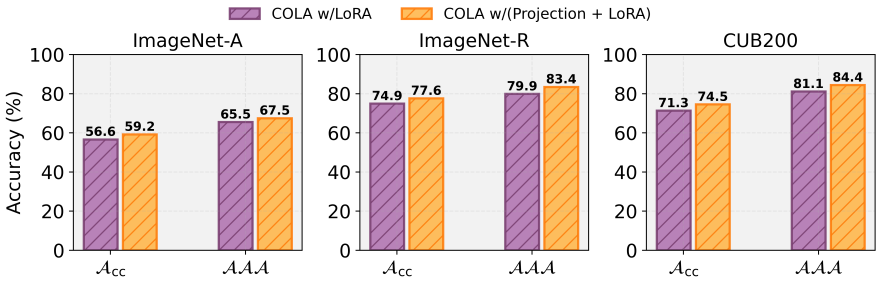

Fig. 3: Ablation study on the soft orthogonal projection matrix $\mathbf{P}$ in COLA across ImageNet-A, ImageNet-R, and CUB200 over 10 tasks, demonstrating consistent improvements in $\mathcal{A}_{cc}$ and $\mathcal{AAA}$ metrics when $\mathbf{P}$ is incorporated.

Table 5: Ablation study on attention projection selection comparing Q/V and K/V LoRA placements across three benchmark datasets at task sequences T=10 and T=20.

Task	Projections	ImageNet-R (%)		ImageNet-A (%)		CUB200 (%)	
		$\mathcal{A}_{cc}$	$\mathcal{AAA}$	$\mathcal{A}_{cc}$	$\mathcal{AAA}$	$\mathcal{A}_{cc}$	$\mathcal{AAA}$
10	K/V	77.13	83.09	57.86	65.99	74.04	82.42
10	Q/V	77.56	83.40	59.20	67.46	74.51	84.38
20	K/V	75.96	81.95	51.33	60.64	68.17	77.08
20	Q/V	76.12	82.27	52.17	61.47	69.18	77.73

after 10 tasks compared to SD-LoRA's 77.04%, demonstrating its resistance to catastrophic forgetting. Notably, COLA's accuracy curve shows fewer fluctuations and smoother degradation patterns compared to existing methods, indicating more stable optimization dynamics.

Computational Efficiency and Memory Footprint. Table 4 presents the computational efficiency and memory footprint of COLA compared to recent continual learning methods. COLA achieves a forward-pass cost of 32.22 GFLOPs, substantially lower than LoRA-based methods such as InfLoRA and SD-LoRA (35.12 GFLOPs) and prompt-based methods such as L2P and HiDe-Prompt (70 GFLOPs). Despite its reduced computational cost, COLA maintains a competitive learnable parameter count of 0.37M, comparable to InfLoRA and lower than L2P (0.48M), while requiring no stored features, unlike methods such as HiDe-Prompt. Together, the lower GFLOPs, minimal parameter overhead, and minimal additional memory storage demonstrate that COLA is a significantly more efficient and scalable solution for continual learning.

6 Ablation Study

To assess the contribution of each component in COLA, we conduct a series of ablation studies. Fig. 3 empirically validates the critical role of soft orthogonal projection in COLA by comparing the full framework against LoRA-only adaptation. The results demonstrate consistent performance gains attributed to the

Table 6: Sensitivity analysis of COLA with respect to the residual scaling factor (γ) initialization values. The model attains peak accuracy at $\gamma = 0.20$, highlighting the importance of balanced covariance guidance. Results are reported as the mean over four independent runs.

γ	ImageNet-A (%)		ImageNet-R (%)		CUB200 (%)	
	$\mathcal{A}_{cc}$	$\mathcal{AAA}$	$\mathcal{A}_{cc}$	$\mathcal{AAA}$	$\mathcal{A}_{cc}$	$\mathcal{AAA}$
0.08	$56.55_{\pm 2.15}$	$65.95_{\pm 0.65}$	$76.87_{\pm 0.24}$	$81.44_{\pm 0.27}$	$73.75_{\pm 0.86}$	$82.10_{\pm 0.57}$
0.10	$57.93_{\pm 1.35}$	$66.06_{\pm 0.78}$	$76.98_{\pm 0.44}$	$82.12_{\pm 0.32}$	$74.02_{\pm 0.84}$	$83.10_{\pm 0.97}$
0.20	$59.20_{\pm 0.87}$	$\mathbf{67.46_{\pm 0.58}}$	$\mathbf{77.56_{\pm 0.35}}$	$\mathbf{83.40_{\pm 0.38}}$	$\mathbf{74.51_{\pm 0.88}}$	$\mathbf{84.38_{\pm 0.97}}$
0.40	$\mathbf{58.45_{\pm 1.87}}$	$65.43_{\pm 0.55}$	$77.04_{\pm 0.52}$	$82.10_{\pm 0.26}$	$74.10_{\pm 0.81}$	$83.03_{\pm 0.95}$
0.60	$55.43_{\pm 1.34}$	$62.88_{\pm 0.68}$	$73.23_{\pm 0.54}$	$78.87_{\pm 0.98}$	$69.45_{\pm 1.08}$	$80.12_{\pm 1.02}$

projection matrix, with an increment of 2.6% on ImageNet-A, 2.7% on ImageNet-R, and 3.2% on CUB200 across $\mathcal{A}_{cc}$ metrics. These results show that standard LoRA remains vulnerable to task interference as new adaptations implicitly overwrite feature subspaces occupied by previous tasks. The covariance-guided projection mechanism addresses this limitation by enforcing geometric independence between task-specific representations, projecting new task features into subspaces orthogonal to the consolidated global covariance structure that encodes prior knowledge. The consistency of these improvements across diverse domains validates that soft orthogonal alignment addresses a fundamental challenge in continual learning rather than being dataset-specific, effectively balancing plasticity for new information with stability for retained knowledge. The ablation results show that low-rank adaptation alone is insufficient to prevent catastrophic forgetting, underscoring that the integration of covariance-guided orthogonal projection with LoRA is the primary driver of COLA's superior CL performance.

We further validate the choice of attention projection placement by comparing Q/V and K/V configurations in Table 5 across 10 and 20 task sequences on ImageNet-R, ImageNet-A, and CUB200 datasets. We empirically find that Q/V projections consistently outperform K/V projections across all benchmarks and metrics, confirming that applying LoRA to the query and value projections yields more effective feature adaptation for continual learning. Table 6 examines the influence of the residual scaling parameter γ, which balances stability and plasticity within COLA's adaptation mechanism by controlling the relative contribution between orthogonally projected features and original low-rank representations. Based on robust testing, the 0.20 value achieves the best performance, attaining 67.46% $\mathcal{AAA}$ on ImageNet-A and 84.38% on CUB200. At lower values, the model overemphasizes orthogonal projection at the expense of task-specific feature expressiveness, reducing the average accuracy while maintaining reasonable $\mathcal{AAA}$ scores. Conversely, excessively high values ($\gamma = 0.60$) substantially degrade performance across all metrics, with ImageNet-R dropping $\mathcal{AAA}$ to 78.87%, as the model prioritizes unconstrained low-rank features over orthogonal guidance, leading to increased task interference and catastrophic forgetting.

Architectural Generalizability. Table 7 demonstrates COLA's architectural generalizability across diverse Vision Transformer backbones, validating its

Table 7: Architectural scalability evaluation of COLA using different backbones across three benchmarks to assess generalization across varying model capacities. Results report $\mathcal{A}_{cc}$ (%), $\mathcal{AAA}$ (%), forgetting rate ($\mathcal{F}$%), inference time, and peak GPU memory consumption with a batch size of 64 over 10 tasks.

Dataset	Model Type	$\mathcal{A}_{cc}$	$\mathcal{AAA}$	$\mathcal{F}$	Inference (ms)	Memory (GB)
ImageNet-A [9]	ViT-B/16	59.20	67.46	11.83	11.36	10.77
	DeiT-B/16	29.66	37.72	13.62	11.37	10.77
	DeiT-S/16	13.12	20.76	11.38	10.08	5.28
ImageNet-R [2]	ViT-B/16	77.56	83.40	9.02	11.37	10.77
	DeiT-B/16	71.92	76.70	7.57	11.37	10.77
	DeiT-S/16	53.92	62.93	13.62	10.08	5.28
CUB200 [30]	ViT-B/16	74.51	84.38	11.96	11.37	10.77
	DeiT-B/16	65.22	73.76	14.44	11.38	10.77
	DeiT-S/16	50.98	61.53	16.62	10.07	5.28

scalability to different model capacities. COLA maintains strong performance with ViT-B/16, achieving 59.20% $\mathcal{A}_{cc}$ on ImageNet-A, 77.56% on ImageNet-R, and 74.51% on CUB200, with a consistent computational footprint of 11.36-11.37ms inference time and 10.77GB peak memory consumption. COLA demonstrates consistent scalability across alternative architectures, with DeiT-B/16 achieving competitive performance at comparable computational overhead. Notably, COLA remains effective even with the lightweight DeiT-S/16, delivering 53.92% on ImageNet-R and 50.98% on CUB200 while requiring only 5.28GB memory and 10.08ms inference time. The performance gap between DeiT-S/16 and ViT-B/16 is attributed to DeiT-S/16's smaller model capacity and embedding dimension relative to ViT-B/16, which constrains its ability to capture complex feature distributions in challenging CL scenarios. Despite these capacity constraints, COLA maintains reasonable forgetting rates of 9.02% on ImageNet-R and 11.96% on CUB200 with ViT-B/16, demonstrating that the covariance-guided orthogonal projection mechanism effectively mitigates catastrophic forgetting across architectural scales. These results collectively confirm that COLA maintains efficient and reliable CL performance even under varying computational constraints. Additional ablation studies on model scalability and hyperparameter sensitivity are provided in the *supplementary document*.

7 Conclusion

In this work, we introduced COLA, a rehearsal-free continual learning framework that integrates low-rank adaptation with covariance-guided orthogonal subspace alignment. Unlike prior methods, COLA incrementally identifies and stabilizes orthogonal representational subspaces, ensuring that new knowledge is integrated without interfering with previously learned tasks. By dynamically consolidating task-specific information into a shared low-rank representation, COLA empirically mitigates catastrophic forgetting and maintains stable adaptation without increasing the parameter count or memory overhead. As a result, COLA outperforms existing prompt and LoRA-based methods across diverse class-incremental

benchmarks, attaining a 4.8% ($\mathcal{AAA}$) improvement on task sequences of up to 25 tasks. Furthermore, COLA is evaluated across diverse datasets and multiple backbone architectures to validate its generalizability, demonstrating consistent performance across varying experimental settings. These results confirm that COLA not only preserves past knowledge but also scales robustly to long task sequences.

In future work, we aim to extend COLA to vision-language models such as CLIP and evaluate it on challenging real-world streams with Gaussian task patterns, assessing its adaptability and scalability in dynamic settings.

Acknowledgements

This research is partially supported by the DST–NSF Project (Project No. DST/INT/USA/NSF-DST/Debasis/P-13/2024) and the TIH IoT and IoE Project (Project ID: TDP06-A-26). The authors gratefully acknowledge the financial support provided by these projects.

References

1. Adhikari, S., Chandra, D.S., Srijith, P., Wasnik, P., Oneo, N.: Adaprefix++: Integrating adapters, prefixes and hypernetwork for continual learning. In: 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 7298–7307. IEEE (2025) 4
2. Boschini, M., Bonicelli, L., Porrello, A., Bellitto, G., Pennisi, M., Palazzo, S., Spampinato, C., Calderara, S.: Transfer without forgetting. In: European conference on computer vision. pp. 692–709. Springer (2022) 9, 14
3. Cheng, Q., Wan, Y., Wu, L., Hou, C., Zhang, L.: Continuous subspace optimization for continual learning. arXiv preprint arXiv:2505.11816 (2025) 3, 4, 10, 11
4. Dai, Y., Hong, X., Wang, Y., Ma, Z., Jiang, D., Wang, Y.: Prompt customization for continual learning. IEEE Transactions on Artificial Intelligence (2025) 4
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009) 9
6. Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., Hu, S., Chen, Y., Chan, C.M., Chen, W., et al.: Parameter-efficient fine-tuning of large-scale pre-trained language models. Nature machine intelligence **5**(3), 220–235 (2023) 4
7. Dosovitskiy, A.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 9
8. Farajtabar, M., Azizan, N., Mott, A., Li, A.: Orthogonal gradient descent for continual learning. In: International conference on artificial intelligence and statistics. pp. 3762–3773. PMLR (2020) 3, 4, 8
9. Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., Song, D.: Natural adversarial examples. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15262–15271 (2021) 9, 14
10. Hu, E.J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022) 2, 4

11. Hu, Z., Li, Y., Lyu, J., Gao, D., Vasconcelos, N.: Dense network expansion for class incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11858–11867 (2023) 4
12. Huang, W.C., Chen, C.F., Hsu, H.: Ovor: Oneprompt with virtual outlier regularization for rehearsal-free class-incremental learning. arXiv preprint arXiv:2402.04129 (2024) 9
13. Jia, M., Tang, L., Chen, B.C., Cardie, C., Belongie, S., Hariharan, B., Lim, S.N.: Visual prompt tuning. In: European conference on computer vision. pp. 709–727. Springer (2022) 4
14. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al.: Highly accurate protein structure prediction with alphafold. nature **596**(7873), 583–589 (2021) 1
15. Kato, I., Hotta, K.: Accuracy improvement of prompt-based continual learning with past information. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5144–5152 (2025) 4
16. Kim, D., Han, B.: On the stability-plasticity dilemma of class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20196–20204 (2023) 2
17. Li, Z., Zhao, L., Zhang, Z., Zhang, H., Liu, D., Liu, T., Metaxas, D.N.: Steering prototypes with prompt-tuning for rehearsal-free continual learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2523–2533 (2024) 2
18. Liang, Y.S., Li, W.J.: Inflora: Interference-free low-rank adaptation for continual learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23638–23647 (2024) 2, 3, 4, 9, 10, 11
19. Liu, Y., Schiele, B., Sun, Q.: Rmm: Reinforced memory management for class-incremental learning. Advances in neural information processing systems **34**, 3478–3490 (2021) 4
20. Lu, Y., Zhang, S., Cheng, D., Xing, Y., Wang, N., Wang, P., Zhang, Y.: Visual prompt tuning in null space for continual learning. Advances in neural information processing systems **37**, 7878–7901 (2024) 4, 10
21. Luo, Z., Liu, Y., Schiele, B., Sun, Q.: Class-incremental exemplar compression for class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11371–11380 (2023) 4
22. McDonnell, M.D., Gong, D., Parvaneh, A., Abbasnejad, E., Van den Hengel, A.: Ranpac: Random projections and pre-trained models for continual learning. Advances in Neural Information Processing Systems **36**, 12022–12053 (2023) 3, 10
23. Oja, E.: Simplified neuron model as a principal component analyzer. Journal of mathematical biology **15**(3), 267–273 (1982) 8
24. Robins, A.: Catastrophic forgetting, rehearsal and pseudorehearsal. Connection Science **7**(2), 123–146 (1995) 1
25. Rolnick, D., Ahuja, A., Schwarz, J., Lillicrap, T., Wayne, G.: Experience replay for continual learning. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019) 4
26. Saha, G., Garg, I., Roy, K.: Gradient projection memory for continual learning. In: International Conference on Learning Representations (2021) 3, 4, 8
27. Smith, J., Hsu, Y.C., Balloch, J., Shen, Y., Jin, H., Kira, Z.: Always be dreaming: A new approach for data-free class-incremental learning. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9374–9384 (2021) 4

28. Smith, J.S., Karlinsky, L., Gutta, V., Cascante-Bonilla, P., Kim, D., Arbelle, A., Panda, R., Feris, R., Kira, Z.: 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. pp. 11909–11919 (2023) 2, 3, 4, 10, 11
29. Sun, Z., Mu, Y., Hua, G.: Regularizing second-order influences for continual learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20166–20175 (2023) 4
30. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The caltech-ucsd birds-200-2011 dataset (2011) 9, 14
31. Wang, F.Y., Zhou, D.W., Ye, H.J., Zhan, D.C.: Foster: Feature boosting and compression for class-incremental learning. In: European conference on computer vision. pp. 398–414. Springer (2022) 4
32. Wang, H., Lu, H., Yao, L., Gong, D.: Self-expansion of pre-trained models with mixture of adapters for continual learning. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 10087–10098 (2025) 4
33. Wang, L., Xie, J., Zhang, X., Huang, M., Su, H., Zhu, J.: Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. Advances in Neural Information Processing Systems **36**, 69054–69076 (2023) 2, 3, 4, 10, 11
34. Wang, X., Zhuang, Z., Zhang, Y.: Plan: Proactive low-rank allocation for continual learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2909–2918 (2025) 3, 10, 11
35. Wang, Y., Huang, Z., Hong, X.: S-prompts learning with pre-trained transformers: An occam's razor for domain incremental learning. Advances in Neural Information Processing Systems **35**, 5682–5695 (2022) 1
36. Wang, Z., Zhang, Z., Ebrahimi, S., Sun, R., Zhang, H., Lee, C.Y., Ren, X., Su, G., Perot, V., Dy, J., et al.: Dualprompt: Complementary prompting for rehearsal-free continual learning. In: European conference on computer vision. pp. 631–648. Springer (2022) 2, 3, 4, 10, 11
37. Wang, Z., Zhang, Z., Lee, C.Y., Zhang, H., Sun, R., Ren, X., Su, G., Perot, V., Dy, J., Pfister, T.: Learning to prompt for continual learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 139–149 (2022) 2, 3, 4, 9, 10, 11
38. Wu, Y., Piao, H., Huang, L.K., Wang, R., Li, W., Pfister, H., Meng, D., Ma, K., Wei, Y.: Sd-lora: Scalable decoupled low-rank adaptation for class incremental learning. arXiv preprint arXiv:2501.13198 (2025) 2, 3, 5, 9, 10, 11
39. Yang, S.H., Oikarinen, T., Weng, T.W.: Concept-driven continual learning. Transactions on Machine Learning Research (2024) 4
40. Yin, H., Feng, T., Lyu, F., Shang, F., Liu, H., Feng, W., Wan, L.: Beyond background shift: Rethinking instance replay in continual semantic segmentation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 9839–9848 (2025) 4
41. Zhao, L., Zhang, X., Yan, K., Ding, S., Huang, W.: Safe: Slow and fast parameter-efficient tuning for continual learning with pre-trained models. Advances in Neural Information Processing Systems **37**, 113772–113796 (2024) 2
42. Zhou, D.W., Cai, Z.W., Ye, H.J., Zhang, L., Zhan, D.C.: Dual consolidation for pre-trained model-based domain-incremental learning. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 20547–20557 (2025) 2
43. Zhou, D.W., Zhang, Y., Wang, Y., Ning, J., Ye, H.J., Zhan, D.C., Liu, Z.: Learning without forgetting for vision-language models. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025) 1, 2