RE-LoRA: Residual Evolving LoRA for Continual Learning

Anonymous authors
Paper under double-blind review

Abstract

Rehearsal-free class-incremental learning (CIL) with frozen pre-trained models is commonly realized through Low-Rank Adaptation (LoRA), yet existing methods still face a fundamental stability-plasticity-efficiency dilemma: they either allocate task-specific low-rank components whose parameter footprint grows with the number of tasks, or keep a compact shared adapter and rely on a regularization loss term to reduce inter-task interference. We propose Residual Evolving LoRA (RE-LoRA), a rehearsal-free framework built upon a single evolving LoRA that attains strong stability and sustained plasticity under a constant parameter budget. RE-LoRA comprises three components. Residual Adapter Learning (RAL) trains each task's low-rank update as a residual that is additively composed with the frozen consolidated adapter in weight space, so that a new task refines rather than overwrites historical knowledge without any auxiliary regularization. Historical Knowledge Consolidation (HKC) then merges the residual update into the historical adapter and re-compresses the result via truncated SVD, which yields the optimal low-rank approximation under a fixed rank budget, so that the single adapter evolves across tasks while its parameter and memory footprints remain constant. Finally, Prototype-Guided Task Routing (PGTR) uses stable prototypes from the frozen backbone to identify the task and masks the logits to its classes, eliminating prediction bias in inference. We prove that HKC achieves the optimal approximation of the RAL combined adaptation in the Frobenius norm under a given rank constraint, and that its truncation error grows at most linearly without compounding over the task stream. Extensive experiments on CIFAR-100, DomainNet, ImageNet-R, and ImageNet-A across ViT-B/16, DINO, and iBOT backbones show that RE-LoRA consistently outperforms state-of-the-art LoRA-based CIL baselines on every evaluated dataset.

1 Introduction

Leveraging large-scale pre-trained models (PTMs) has emerged as a powerful paradigm for rehearsal-free class-incremental learning (CIL) (van de Ven & Tolias (2018)). Rather than continually modifying the full backbone, a frozen PTM can be adapted through a small set of trainable parameters, following the paradigm of parameter-efficient fine-tuning (PEFT). Among existing PEFT techniques, Low-Rank Adaptation (LoRA) (Hu et al. (2022)) is particularly attractive. Specifically, the LoRA-based method parameterizes each weight update as the product of two low-rank matrices, substantially reducing the number of trainable parameters while retaining the adaptation capacity of the original model. This property has motivated a growing body of LoRA-based continual learning methods (He et al. (2025); Wu et al. (2025); Liang & Li (2024); Qiu et al. (2025); Lai et al.; Zheng et al. (2026)), which have demonstrated strong performance on CIL benchmarks.

Despite these advances, LoRA-based continual learning still faces a fundamental *stability–plasticity–efficiency* dilemma, where the stability and plasticity denote the abilities of preserving old knowledge and adapting to new tasks. To address these challenges, one popular approach preserves historical knowledge by allocating task-specific low-rank components. SD-LoRA (Wu et al. (2025)), for example, decouples magnitude and direction across tasks, while CL-LoRA (He et al. (2025)) augments a shared adapter with task-specific adapters. Although effective at preserving previously learned knowledge, these methods retain additional components for every encountered task, so the parameter footprint grows with the task sequence and long-term scalability is limited.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

The second direction instead maintains a compact shared LoRA and mitigates forgetting through explicit regularization. EWC-LoRA (Zheng et al. (2026)) penalizes changes to parameters deemed important by Fisher information, while MoCL (Lai et al.) employs curvature-aware gating to suppress updates that conflict with historical knowledge. Such approaches keep the parameter count fixed, but they introduce additional computation and memory overhead. Moreover, as historical constraints accumulate, they increasingly restrict the model's capacity to adapt to new tasks. These limitations motivate a fundamental question: *can rehearsal-free CIL achieve strong historical stability and sustained plasticity while keeping both the model parameter budget and the optimization objective unchanged?*

We answer this question with our proposed **R**esidual **E**volving **LoRA** (**RE-LoRA**), a rehearsal-free framework built upon a single evolving LoRA that attains strong stability and sustained plasticity without expanding the parameter count or introducing auxiliary regularization. RE-LoRA comprises three complementary components. (i) **Residual Adapter Learning (RAL)** trains each task's low-rank update as a residual that is additively composed with the frozen consolidated adapter in weight space, so that a new task refines rather than overwrites the knowledge of previous ones, without any auxiliary regularization term. (ii) **Historical Knowledge Consolidation (HKC)** then merges the residual update into the historical adapter and re-compresses the result via truncated SVD to the fixed rank budget, yielding the optimal low-rank approximation under that budget. The single adapter is thereby re-parameterized and *evolves* across tasks, while its parameter count remains constant in size. (iii) **Prototype-Guided Task Routing (PGTR)** eliminates prediction bias at inference time. It uses stable prototypes from the frozen backbone to identify the most likely task for each test sample, then applies a logit mask that restricts the classifier to that task's classes, yielding interference-free predictions with only one additional forward pass through the frozen backbone.

Our contributions are summarized as follows. (1) We propose RE-LoRA, a LoRA-based CL framework that couples RAL and HKC to jointly achieve strong stability, sustained plasticity, and constant parameter overhead. (2) We prove that HKC's merge-and-compress consolidation is optimal for retaining the RAL consolidated knowledge under a bounded rank budget, and that its truncation error does not compound over the task stream but instead accumulates at most linearly. (3) We introduce prototype-guided task routing, which decouples task identification from task-specific feature adaptation and eliminates prediction bias at inference time. (4) Extensive experiments on CIFAR-100, DomainNet, ImageNet-R, and ImageNet-A under diverse task schedules and backbone architectures (including ViT-B/16, DINO, and iBOT) demonstrate that RE-LoRA consistently outperforms all state-of-the-art LoRA-based CL baselines on every evaluated dataset.

2 METHODOLOGY

2.1 PRELIMINARIES

Problem Formulation. We consider the standard class-incremental learning (CIL) setting, in which a model is trained over a sequence of N disjoint tasks $\{\mathcal{T}_1, \ldots, \mathcal{T}_N\}$. Each task $\mathcal{T}_t$ provides a training set $\mathcal{D}_t^{\mathrm{tr}} = \{(\mathbf{x}_i^{\mathrm{tr}}, \mathbf{y}_i^{\mathrm{tr}})\}_{i=1}^{n_t^{\mathrm{tr}}}$ and testing set $\mathcal{D}_t^{\mathrm{te}} = \{(\mathbf{x}_i^{\mathrm{te}}, \mathbf{y}_i^{\mathrm{te}})\}_{i=1}^{n_t^{\mathrm{te}}}$ consisting of samples from a set of newly introduced classes. During the training of $\mathcal{T}_t$, the model has access only to $\mathcal{D}_t^{\mathrm{tr}}$ while all preceding datasets $\{\mathcal{D}_1^{\mathrm{tr}}, \ldots, \mathcal{D}_{t-1}^{\mathrm{tr}}\}$ are strictly unavailable. At inference time, the model must classify samples from the cumulative set $\bigcup_{j=1}^{t} \mathcal{D}_j^{\mathrm{te}}$ without access to task identifiers. Nowadays, the core challenge in CIL no longer lies merely in balancing **stability** on historical classes and **plasticity** for new ones, but in satisfying both under parameter and computational constraints (**efficiency**).

Low-Rank Adaptation (LoRA). PEFT via LoRA (Hu et al. (2022)) minimizes the number of trainable parameters while preserving the representational capacity of a pre-trained backbone. Consider an L-layer Vision Transformer (ViT) with frozen pre-trained weights $\Omega = \{\mathbf{W}_0^{(\ell)}\}_{\ell=1}^{L}$. For the ℓ-th linear projection $\mathbf{W}_0^{(\ell)} \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{in}}}$, LoRA constrains the update $\Delta\mathbf{W}^{(\ell)}$ to a low-rank factorization $\Delta\mathbf{W}^{(\ell)} = \mathbf{B}^{(\ell)}\mathbf{A}^{(\ell)}$, where $\mathbf{A}^{(\ell)} \in \mathbb{R}^{r_{\mathrm{base}} \times d_{\mathrm{in}}}$ and $\mathbf{B}^{(\ell)} \in \mathbb{R}^{d_{\mathrm{out}} \times r_{\mathrm{base}}}$ are trainable down- and up-projection matrices, respectively, with rank $r_{\mathrm{base}} \ll \min\{d_{\mathrm{out}}, d_{\mathrm{in}}\}$. d_{in} and d_{out} are the input and output dimensions, respectively. The forward pass is modified as $\mathbf{f}^{(\ell)} = \mathbf{W}_0^{(\ell)}\mathbf{x}^{(\ell)} + \mathbf{B}^{(\ell)}\mathbf{A}^{(\ell)}\mathbf{x}^{(\ell)}$.

2.2 RESIDUAL ADAPTER LEARNING

RE-LoRA maintains a single evolving adapter state $\Delta\mathbf{W}_{\mathrm{hist}}^{(\ell)}$ per layer that is carried across the task sequence. In this section, we propose **Residual Adapter Learning (RAL)**, which specifies *how a novel task is learned* given that state. *How the state itself is updated* is deferred to Sec. 2.3.

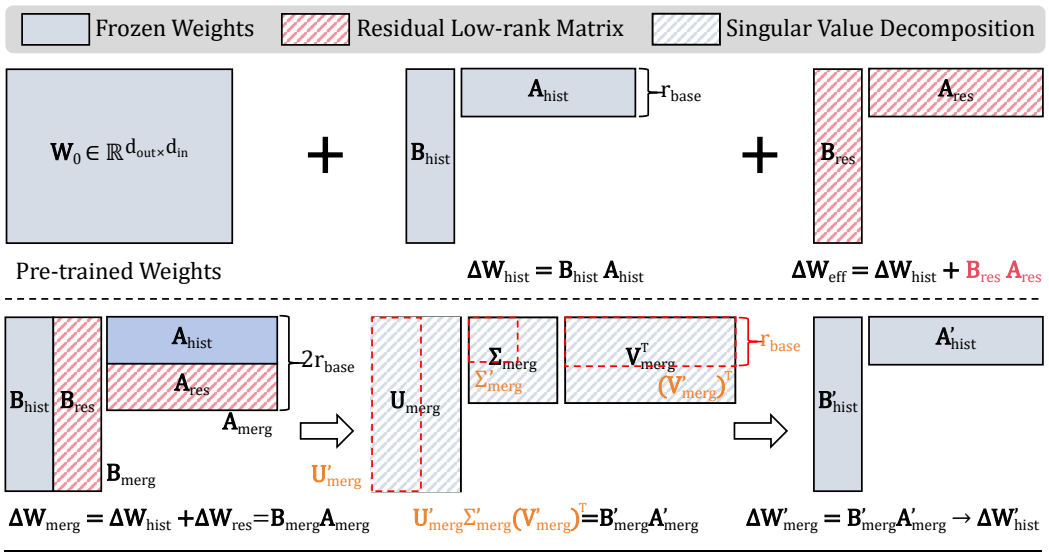


Figure 1: Overview of RE-LoRA's learning and weight consolidation. Each layer maintains a frozen historical adapter $\Delta \mathbf{W}_{\text{hist}} = \mathbf{B}_{\text{hist}} \mathbf{A}_{\text{hist}}$. During each task's training, it also creates a trainable residual adapter $\Delta \mathbf{W}_{\text{res}} = \mathbf{B}_{\text{res}} \mathbf{A}_{\text{res}}$. Their sum forms the effective update $\Delta \mathbf{W}_{\text{eff}} = \Delta \mathbf{W}_{\text{hist}} + \Delta \mathbf{W}_{\text{res}}$ defined in Eq. (1), which is added to the frozen weight $\mathbf{W}_0$ during forward passes; only the residual factors receive gradients, so historical knowledge is never overwritten during learning. After each task, the two branches are first merged into $\Delta \mathbf{W}_{\text{merg}}$ of rank at most $2r_{\text{base}}$ by simple matrix addition. Truncated SVD then retains its top-r_{base} components, which are re-factorized into $\mathbf{B}'_{\text{merg}}$ and $\mathbf{A}'_{\text{merg}}$ (Eq. (3)). The result is frozen as the new $\Delta \mathbf{W}'_{\text{hist}}$, and the residual adapter is discarded. Thus, the residual branch exists only during training, while a single adapter of constant rank r_{base} evolves over the task stream.

At the beginning of each task $\mathcal{T}_t$, we initialize a fresh LoRA module with parameters $\mathbf{A}_{\text{res}}^{(\ell)}$ and $\mathbf{B}_{\text{res}}^{(\ell)}$ of rank r_{base}. The adapter state $\Delta \mathbf{W}_{\text{hist}}^{(\ell)}$ is inherited from the previous task $\mathcal{T}_{t-1}$ and is kept frozen throughout the current task. The current task is then trained with the effective weight update

$$\Delta \mathbf{W}_{\text{eff}}^{(\ell)} = \Delta \mathbf{W}_{\text{hist}}^{(\ell)} + \mathbf{B}_{\text{res}}^{(\ell)} \mathbf{A}_{\text{res}}^{(\ell)}, \tag{1}$$

which augments the frozen historical knowledge with a residual adaptation for the new classes. For the first task $\mathcal{T}_1$ we set $\Delta \mathbf{W}_{\text{hist}}^{(\ell)} = \mathbf{0}$, thus its effective weight update reduces to a vanilla LoRA. The forward pass of layer ℓ is accordingly

$$\mathbf{f}^{(\ell)} = \mathbf{W}_0^{(\ell)} \mathbf{x}^{(\ell)} + \Delta \mathbf{W}_{\text{eff}}^{(\ell)} \mathbf{x}^{(\ell)}. \tag{2}$$

Since the two contributions are composed additively in weight space, the historical adapter is never overwritten while training a new task. Only $\mathbf{A}_{\text{res}}$ and $\mathbf{B}_{\text{res}}$ receive gradients, and the optimization objective for $\mathcal{T}_t$ is simply the cross-entropy loss $\mathcal{L}_{\text{CE}}$ evaluated on the effective forward pass of Eq. (2), without any auxiliary regularization term or explicit gradient projection.

2.3 Historical Knowledge Consolidation

At the end of each task, the adapter state must incorporate the residual adaptation just learned so as to serve as the starting point for the next task. This update is performed by **Historical Knowledge Consolidation (HKC)**, which merges the historical and residual adapters and re-compresses the merged result to the same rank r_{base}, thereby strictly respecting the parameter budget.

For the first task $\mathcal{T}_1$, there is no historical knowledge to preserve. Since $\Delta \mathbf{W}_{\text{hist}} = \mathbf{0}$, the residual adapter trained on $\mathcal{T}_1$ is already of rank r_{base}, and the consolidation simply freezes its factors as the initial adapter state for $\mathcal{T}_2$. For every subsequent task, we merge the residual adapter $\Delta \mathbf{W}_{\text{res}} = \mathbf{B}_{\text{res}} \mathbf{A}_{\text{res}}$ with the historical adapter $\Delta \mathbf{W}_{\text{hist}}$ as follows.

Weight Merging and Compression. First, $\Delta \mathbf{W}_{\text{hist}}$ and $\Delta \mathbf{W}_{\text{res}}$ are merged into a single dense matrix $\Delta \mathbf{W}_{\text{merg}} = \Delta \mathbf{W}_{\text{hist}} + \Delta \mathbf{W}_{\text{res}} = \mathbf{B}_{\text{merg}} \mathbf{A}_{\text{merg}}$. This operation temporarily breaks

the low-rank factorization to allow a global spectral analysis. To restore the low-rank structure and eliminate redundancy, we compute the singular value decomposition (SVD) of $\Delta\mathbf{W}_{\text{merg}}$: $\Delta\mathbf{W}_{\text{merg}} = \mathbf{U}_{\text{merg}}\mathbf{\Sigma}_{\text{merg}}\mathbf{V}_{\text{merg}}^{\top}$, where $\mathbf{\Sigma}_{\text{merg}} = \text{diag}(\sigma_1, \ldots, \sigma_m)$ and $\mathbf{U}_{\text{merg}} \in \mathbb{R}^{d_{\text{out}} \times m}$ and $\mathbf{V}_{\text{merg}} \in \mathbb{R}^{d_{\text{in}} \times m}$ have orthonormal columns. Here $\sigma_1 \geq \cdots \geq \sigma_m \geq 0$ are the singular values of $\Delta\mathbf{W}_{\text{merg}}$, and $m = \min(d_{\text{out}}, d_{\text{in}})$. Then, we apply truncation to $\mathbf{\Sigma}_{\text{merg}}$, retaining the top-$r_{\text{base}}$ singular values. Let $\mathbf{\Sigma}'_{\text{merg}}$ contain these singular values, and let $\mathbf{U}'_{\text{merg}}$ and $\mathbf{V}'_{\text{merg}}$ be the corresponding singular vectors. The merged weight is then reconstructed in low-rank form as

$$\mathbf{A}'_{\text{merg}} = \sqrt{\mathbf{\Sigma}'_{\text{merg}}}(\mathbf{V}'_{\text{merg}})^{\top}, \qquad \mathbf{B}'_{\text{merg}} = \mathbf{U}'_{\text{merg}}\sqrt{\mathbf{\Sigma}'_{\text{merg}}}, \tag{3}$$

so that $\Delta\mathbf{W}'_{\text{merg}} = \mathbf{B}'_{\text{merg}}\mathbf{A}'_{\text{merg}}$ retains only the top-r_{base} components. The current residual adapter $\Delta\mathbf{W}_{\text{res}}$ is discarded, and $\Delta\mathbf{W}_{\text{hist}}$ is updated to $\Delta\mathbf{W}'_{\text{hist}} = \Delta\mathbf{W}'_{\text{merg}}$ for the next task. Since the rank budget r_{base} is fixed, the network parameter number remains constant irrespective of the sequence length, while the knowledge of historical tasks is folded into the same evolving adapter.

2.4 Prototype-Guided Task Routing

While RAL and HKC jointly guarantee that historical knowledge is preserved at the feature level, a critical issue remains at inference level: **prediction bias toward newly learned classes**. Specifically, passing a test sample through the latest model can cause the classifier to produce anomalously high logits for new-task classes, even when the sample belongs to a historical task. To solve this problem, existing methods (Zhang et al. (2023); Wang et al. (2025); Liang & Li (2024)) often perform classifier alignment with auxiliary regularization terms. However, this technique adds a loss to balance during training and can impair classification ability on new task. To eliminate prediction bias without any regularization on the classifier, we introduce a **Prototype-Guided Task Routing (PGTR)** mechanism that operates purely at prediction time. For a given test sample with unknown task identity, PGTR first predicts its original task and then restricts the logits comparison to classes within that task. This is achieved by maintaining a stable, task-agnostic reference frame using the frozen pre-trained backbone, thus decouples task identification from task-specific feature adaptation.

Stable Prototype Construction. During the training of $\mathcal{T}_t$, we extract features using the frozen backbone without any LoRA adapters. For a class c introduced in $\mathcal{T}_t$, its prototype $\mathbf{p}_c \in \mathbb{R}^{d_{\text{out}}}$ is defined as the empirical mean of its features over all training samples belonging to that class:

$$\mathbf{p}_c = \frac{1}{N_c}\sum\nolimits_{(\mathbf{x}_i^{\text{tr}}, y_i^{\text{tr}}) \in \mathcal{D}_t^{\text{tr}},\, y_i^{\text{tr}} = c} F_{\Omega}(\mathbf{x}_i), \tag{4}$$

where F_{Ω} denotes the frozen backbone, $\mathcal{D}_t^{\text{tr}}$ is the training set of task $\mathcal{T}_t$, and N_c is the number of samples belonging to class c. Because F_{Ω} is strictly frozen, the resulting global prototype matrix $\mathbf{P} = [\mathbf{p}_1, \ldots, \mathbf{p}_{C_{\text{seen}}}]^{\top}$ serves as a stable and unbiased reference frame for all seen C_{seen} classes, remaining completely immune to the feature shifts induced by sequential LoRA adaptations.

Test-Time Routing and Logit Masking. During inference, for a given test sample $\mathbf{x}_i^{\text{te}}$ and the task-shared linear classifier F_{ω}, we first extract its backbone feature $\mathbf{f}_i^{\text{te}} = F_{\Omega}(\mathbf{x}_i^{\text{te}})$. We then compute the cosine similarity between $\mathbf{f}_i^{\text{te}}$ and all stored prototypes to identify the most probable class $\hat{c}$:

$$\hat{c} = \arg\max_{c} \frac{(\mathbf{f}_i^{\text{te}})^{\top}\mathbf{p}_c}{\|\mathbf{f}_i^{\text{te}}\|_2 \|\mathbf{p}_c\|_2}. \tag{5}$$

Since the class prototype matrix extracted from the frozen backbone clearly lacks fine-grained classification capability when directly used as a classifier, we map the predicted class $\hat{c}$ to a specific task index $k = \lfloor \hat{c}/C_{\text{task}} \rfloor$, where C_{task} is the number of classes per task. Finally, we pass $\mathbf{x}_i^{\text{te}}$ through the full model $F_{\Omega'}$ including the current LoRA modules and classifier F_{ω} to obtain the raw logits $\mathbf{z} = F_{\omega}(F_{\Omega'}(\mathbf{x}_i^{\text{te}})), \mathbf{z} \in \mathbb{R}^{C_{\text{total}}}$. To prevent prediction bias, we construct a binary mask $\mathbf{m} \in \{1, 0\}^{C_{\text{total}}}$ and apply it to $\mathbf{z}$:

$$\mathbf{z}'_c = \begin{cases} \mathbf{z}_c, & \text{if } c \in [k \cdot C_{\text{task}}, (k+1) \cdot C_{\text{task}} - 1], \\ -\infty, & \text{otherwise.} \end{cases} \tag{6}$$

The final prediction is obtained by $\hat{y} = \arg\max_c \mathbf{z}'_c$.

This routing strategy effectively addresses the prediction bias problem. By relying on the frozen backbone for task identification, the routing decision is fully decoupled from the task-specific feature

distortions introduced by LoRA. Logit masking then acts as a hard constraint, forcing the classifier to compete only among the classes of the identified task, thereby guaranteeing robust and interference-free predictions at inference time. This requires only an additional forward pass through the frozen backbone and a storage overhead equivalent to one extra classifier head. The pseudocode of RE-LoRA training process is outlined in Algorithm 1 in Appendix A.

2.5 Theoretical Analysis

In this section, we provide a theoretical justification for the optimality of the merge-and-compress strategy in the proposed **HKC**, and how its error accumulates over the task stream.

Proposition 2.1 (Optimality of Merge-and-Compress) *Let $\Delta\mathbf{W}_{\text{hist}}$ and $\Delta\mathbf{W}_{\text{res}}$ be the historical and residual adaptation matrices, and let $\Delta\mathbf{W}_{\text{merg}} = \Delta\mathbf{W}_{\text{hist}} + \Delta\mathbf{W}_{\text{res}}$ denote their dense sum. For any prescribed rank budget r_{base}, the proposed strategy of forming $\Delta\mathbf{W}_{\text{merg}}$ and applying truncated SVD (Eq. (3)) yields the optimal low-rank approximation, and it minimizes the reconstruction error of the consolidated adaptation matrix*

$$\Delta\mathbf{W}'_{\text{hist}} = \arg\min_{\text{rank}(\mathbf{W})\le r_{\text{base}}} \|\Delta\mathbf{W}_{\text{hist}} + \Delta\mathbf{W}_{\text{res}} - \mathbf{W}\|_F^2 . \tag{7}$$

Proof. The objective is to find a matrix $\mathbf{W}$ of rank at most r_{base} that best approximates the joint adaptation matrix $\Delta\mathbf{W}_{\text{merg}} = \Delta\mathbf{W}_{\text{hist}} + \Delta\mathbf{W}_{\text{res}}$ in the Frobenius norm. By the classical Eckart–Young theorem (Eckart & Young (1936)), the best rank-k approximation to any matrix $\mathbf{M}$ in the Frobenius (or spectral) norm is given by its truncated SVD retaining the top-k singular values. By computing the full SVD of the dense matrix $\Delta\mathbf{W}_{\text{merg}}$ and truncating it to r_{base} components (the merge-and-compress step of Sec. 2.3), our method exactly realizes this optimal projection.

Remark. Proposition 2.1 shows that merge-and-compress realizes the optimal rank-r_{base} approximation of the combined adaptation, which also explains why two natural alternatives are inferior. (i) *Direct concatenation.* Keeping the historical and residual branches side by side corresponds to the factorization $\Delta\mathbf{W}_{\text{merg}} = [\mathbf{B}_{\text{hist}}, \mathbf{B}_{\text{res}}][\mathbf{A}_{\text{hist}}; \mathbf{A}_{\text{res}}]$, whose rank is up to $t\, r_{\text{base}}$ after t tasks. This degenerates to per-task expansion: the parameter count grows with the sequence and the redundancy among task-specific updates is never removed. (ii) *Direct addition without compression.* Merging the two branches into the dense sum $\Delta\mathbf{W}_{\text{hist}} + \Delta\mathbf{W}_{\text{res}}$ but keeping it uncompressed also violates the budget. Since rank is only subadditive, $\text{rank}(\Delta\mathbf{W}_{\text{hist}} + \Delta\mathbf{W}_{\text{res}})$ can exceed r_{base} even after a single merge and grows toward $\min\{d_{\text{out}}, d_{\text{in}}\}$ as tasks accumulate. In contrast, merging first and then truncating discards exactly the redundant directions and retains maximal information under the strict budget r_{base}.

Strengthening the single-step guarantee above, the following result shows that the compression error does not compound multiplicatively across the task stream. Holding the residual updates fixed, we compare the consolidated adapter with the reference accumulation obtained by summing the same residuals without truncation, and show that their deviation is exactly the sum of the per-step truncation residuals.

Proposition 2.2 (Non-Compounding Error Accumulation) *Let $\mathbf{R}_t = \Delta\mathbf{W}_{\text{res}}^{(t)}$ denote the residual update of task $\mathcal{T}_t$, and let $\mathbf{H}_t = T_{r_{\text{base}}}(\mathbf{H}_{t-1} + \mathbf{R}_t)$, $\mathbf{H}_0 = \mathbf{0}$, be the consolidated adapter produced by the proposed HKC, where $T_r(\cdot)$ returns the rank-r SVD truncation. Let $\mathbf{F}_t = \sum_{s=1}^{t} \mathbf{R}_s$ denote the untruncated reference accumulation formed by summing the residual updates actually generated along the training trajectory, without applying the $T_r(\cdot)$ operator. Then*

$$\mathbf{F}_N - \mathbf{H}_N = \sum_{t=1}^{N} \mathbf{E}_t, \qquad \mathbf{E}_t \triangleq (\mathbf{H}_{t-1} + \mathbf{R}_t) - T_{r_{\text{base}}}(\mathbf{H}_{t-1} + \mathbf{R}_t), \tag{8}$$

and the total deviation satisfies

$$\|\mathbf{F}_N - \mathbf{H}_N\|_F \le \sum_{t=1}^{N} \|\mathbf{E}_t\|_F \le \sqrt{r_{\text{base}}} \sum_{t=1}^{N} \|\mathbf{R}_t\|_F . \tag{9}$$

Proof sketch. The identity follows by induction on t. Since $\mathbf{F}_t = \mathbf{F}_{t-1} + \mathbf{R}_t$ and $\mathbf{H}_t = T_{r_{\text{base}}}(\mathbf{H}_{t-1} + \mathbf{R}_t)$, subtracting the two recursions gives $\mathbf{F}_t - \mathbf{H}_t = (\mathbf{F}_{t-1} - \mathbf{H}_{t-1}) + \mathbf{E}_t$, so

Table 1: Comparison results on CIFAR-100, DomainNet, ImageNet-R, and ImageNet-A (in %). **Bold** and underline indicate the highest and second-highest scores, respectively.

Methods	CIFAR-100 10 tasks		DomainNet 5 tasks		ImageNet-R 10 tasks		ImageNet-A 10 tasks	
	$\bar{A}_{10}(\uparrow)$	Avg.($\uparrow$)	$\bar{A}_{5}(\uparrow)$	Avg.($\uparrow$)	$\bar{A}_{10}(\uparrow)$	Avg.($\uparrow$)	$\bar{A}_{10}(\uparrow)$	Avg.($\uparrow$)
Full Expansion	$98.71_{(0.03)}$	$98.80_{(0.02)}$	$86.75_{(0.05)}$	$87.20_{(0.02)}$	$94.00_{(0.05)}$	$93.77_{(0.04)}$	$78.90_{(0.37)}$	$80.88_{(0.05)}$
Finetune	$79.09_{(1.53)}$	$88.17_{(0.45)}$	$65.57_{(0.20)}$	$75.12_{(0.12)}$	$60.42_{(1.64)}$	$73.18_{(0.32)}$	$32.85_{(1.53)}$	$54.55_{(1.44)}$
L2P	$83.18_{(1.20)}$	$87.69_{(1.05)}$	$70.26_{(0.25)}$	$75.83_{(0.98)}$	$71.26_{(0.44)}$	$76.13_{(0.46)}$	$42.94_{(1.27)}$	$51.40_{(1.95)}$
DualPrompt	$81.48_{(0.86)}$	$86.41_{(0.66)}$	$68.26_{(0.90)}$	$73.84_{(0.45)}$	$68.22_{(0.20)}$	$73.81_{(0.39)}$	$45.49_{(0.96)}$	$54.68_{(1.24)}$
CODA-Prompt	$86.31_{(0.12)}$	$90.67_{(0.22)}$	$70.58_{(0.53)}$	$76.68_{(0.44)}$	$74.05_{(0.41)}$	$78.14_{(0.39)}$	$45.36_{(0.78)}$	$57.03_{(0.94)}$
HiDe-Prompt	$\underline{92.61}_{(0.28)}$	$\underline{94.03}_{(0.01)}$	$72.20_{(0.08)}$	$77.01_{(0.04)}$	$74.65_{(0.14)}$	$78.46_{(0.18)}$	$42.70_{(0.60)}$	$56.32_{(0.40)}$
InfLoRA	$86.34_{(0.76)}$	$91.33_{(0.48)}$	$71.01_{(0.05)}$	$77.75_{(0.03)}$	$74.41_{(0.63)}$	$80.31_{(0.60)}$	$50.75_{(1.33)}$	$64.36_{(1.01)}$
SD-LoRA	$86.77_{(0.30)}$	$90.96_{(0.30)}$	$71.27_{(0.14)}$	$77.70_{(0.07)}$	$72.93_{(2.76)}$	$79.80_{(0.15)}$	$55.23_{(0.94)}$	$66.10_{(0.54)}$
LoRA-P&M	$88.45_{(0.35)}$	$92.89_{(1.13)}$	$\underline{76.91}_{(0.53)}$	$\underline{85.27}_{(0.07)}$	$79.95_{(0.18)}$	$85.29_{(0.93)}$	$56.57_{(0.78)}$	$65.35_{(1.81)}$
CL-LoRA	$87.65_{(0.53)}$	$92.25_{(0.41)}$	$71.06_{(0.31)}$	$77.76_{(0.29)}$	$78.72_{(0.44)}$	$85.20_{(0.45)}$	$57.62_{(0.89)}$	$\underline{70.76}_{(0.63)}$
MoCL	$88.12_{(0.18)}$	$92.74_{(0.14)}$	$74.23_{(0.14)}$	$80.22_{(0.07)}$	$\underline{81.33}_{(0.18)}$	$\underline{86.07}_{(0.19)}$	$58.40_{(0.32)}$	$68.36_{(0.34)}$
BiLoRA	$85.99_{(0.49)}$	$90.62_{(0.42)}$	$69.75_{(0.23)}$	$73.86_{(0.21)}$	$74.28_{(0.92)}$	$77.38_{(0.88)}$	$51.05_{(0.74)}$	$62.82_{(0.65)}$
EWC-LoRA	$87.91_{(0.57)}$	$92.27_{(0.39)}$	$73.46_{(0.16)}$	$79.58_{(0.10)}$	$72.86_{(0.79)}$	$78.95_{(0.86)}$	$\underline{59.89}_{(0.26)}$	$68.33_{(0.67)}$
RE-LoRA	$\mathbf{98.04}_{(0.02)}$	$\mathbf{98.27}_{(0.03)}$	$\mathbf{84.42}_{(0.17)}$	$\mathbf{85.47}_{(0.08)}$	$\mathbf{90.72}_{(0.15)}$	$\mathbf{91.16}_{(0.03)}$	$\mathbf{66.50}_{(1.29)}$	$\mathbf{72.32}_{(0.70)}$

the deviation telescopes into $\mathbf{F}_N - \mathbf{H}_N = \sum_{t=1}^{N} \mathbf{E}_t$. For the norm bound, $\mathbf{M}_t = \mathbf{H}_{t-1} + \mathbf{R}_t$ has rank at most $2r_{\text{base}}$, so the truncated SVD discards only the singular values $\sigma_{r_{\text{base}}+1}, \ldots, \sigma_{2r_{\text{base}}}$; Weyl's inequality (Weyl (1949)) bounds the largest discarded value by $\sigma_1(\mathbf{R}_t) \leq \|\mathbf{R}_t\|_F$, which yields $\|\mathbf{E}_t\|_F \leq \sqrt{r_{\text{base}}}\|\mathbf{R}_t\|_F$, and the triangle inequality over t gives Eq. (9). The complete proof is provided in Appendix B.1.

Corollary 2.3 (Linear Error Growth under Spectral Decay) *If the merged adaptation matrices satisfy a geometric spectral decay $\sigma_i(\mathbf{H}_{t-1} + \mathbf{R}_t) \leq \alpha\rho^{i-1}$ for some $\alpha > 0$, $0 < \rho < 1$ and all t, then*

$$\|\mathbf{E}_t\|_F \leq \frac{\alpha\,\rho^{\,r_{\text{base}}}}{\sqrt{1-\rho^2}}, \qquad \|\mathbf{F}_N - \mathbf{H}_N\|_F \leq \frac{\alpha\,\rho^{\,r_{\text{base}}}}{\sqrt{1-\rho^2}}\,N. \tag{10}$$

Proof sketch. Because $\operatorname{rank}(\mathbf{H}_{t-1} + \mathbf{R}_t) \leq 2r_{\text{base}}$, only the singular values beyond index r_{base} contribute to $\mathbf{E}_t$, and the geometric decay assumption bounds the discarded tail by $\|\mathbf{E}_t\|_F^2 \leq \sum_{i=r_{\text{base}}+1}^{\infty} \alpha^2\rho^{2(i-1)} = \alpha^2\rho^{2r_{\text{base}}}/(1-\rho^2)$. Each per-task truncation error is therefore proportional to $\rho^{r_{\text{base}}}$, and substituting this per-task constant into the non-compounding bound of Proposition 2.2 and summing over the N tasks yields the stated linear growth. Complete proof see Appendix B.2.

Remark. Proposition 2.2 and Corollary 2.3 extend the single-step optimality of Proposition 2.1 to a sequence of tasks. Specifically, let $\mathbf{F}_N = \sum_{t=1}^{N} \mathbf{R}_t$ denote the untruncated reference accumulation formed from the residual updates actually generated along the training trajectory. Proposition 2.2 shows that, for this fixed residual sequence, the deviation between the consolidated adapter $\mathbf{H}_N$ and the reference accumulation $\mathbf{F}_N$ is exactly the sum of the truncation errors introduced at individual consolidation steps. Hence, the result characterizes the cumulative representation error induced by the consolidation operator, without making assumptions about the dynamics by which the residual updates $\mathbf{R}_t$ are learned. Corollary 2.3 further shows that, when the singular values of the pre-truncation merged matrices exhibit geometric decay, each per-task truncation error decreases exponentially with the rank budget r_{base}, while the resulting cumulative bound grows at most linearly with the number of tasks. These results therefore establish that, for a fixed sequence of realized residual updates, repeated low-rank consolidation introduces a controlled and non-compounding truncation error, providing a theoretical characterization of the fidelity of the constant-size RE-LoRA representation with respect to the corresponding untruncated residual accumulation.

3 Experiments

Evaluation Benchmarks. We base our experiments on (Wu et al. (2025); Zheng et al. (2026)) and evaluate RE-LoRA on four standard CL benchmarks: ImageNet-R (split into 5/10/20 tasks) (Hendrycks et al. (2021a)), ImageNet-A (10 tasks) (Hendrycks et al. (2021b)), DomainNet (5 tasks) (Peng et al. (2019)), and CIFAR100 (10 tasks) (Krizhevsky et al. (2009)).

Metrics. Following common practice in CIL, we report the average accuracy after learning task $\mathcal{T}_t$, $\bar{A}_t = \frac{1}{t}\sum_{i=1}^{t} A_{t,i}$, where $A_{t,i}$ denotes the classification accuracy on the i-th task after training on the t-th task, and the overall average accuracy $\text{Avg.} = \frac{1}{N}\sum_{t=1}^{N} \bar{A}_t$, where N is the total number of

Table 2: Stability (↑) and plasticity (↑) scores of different LoRA CL methods, reflecting how well each model retains previous knowledge and adapts to new tasks. We report the normalized form of the two metrics, which is independent of the absolute performance on the dataset.

Methods	CIFAR-100 10 tasks		DomainNet 5 tasks		ImageNet-R 10 tasks		ImageNet-A 10 tasks	
	Stability(↑)	Plasticity(↑)	Stability(↑)	Plasticity(↑)	Stability(↑)	Plasticity(↑)	Stability(↑)	Plasticity(↑)
InfLoRA	$94.84_{(0.52)}$	$95.80_{(1.00)}$	$83.80_{(0.37)}$	$97.34_{(0.29)}$	$92.69_{(0.77)}$	$97.33_{(0.46)}$	$88.63_{(3.39)}$	$72.69_{(3.13)}$
SD-LoRA	$91.85_{(0.60)}$	$98.24_{(0.25)}$	$82.45_{(0.19)}$	$98.69_{(0.15)}$	$91.78_{(0.90)}$	$95.61_{(0.32)}$	$88.61_{(1.05)}$	$92.13_{(2.77)}$
EWC-LoRA	$94.45_{(0.59)}$	$97.99_{(0.50)}$	$91.51_{(0.36)}$	$93.83_{(0.23)}$	$95.62_{(0.42)}$	$93.23_{(0.34)}$	$89.52_{(1.14)}$	$\mathbf{92.78}_{(1.91)}$
RE-LoRA	$\mathbf{99.40}_{(0.10)}$	$\mathbf{99.74}_{(0.02)}$	$\mathbf{97.59}_{(0.23)}$	$\mathbf{99.21}_{(0.03)}$	$\mathbf{98.57}_{(0.12)}$	$\mathbf{97.75}_{(0.21)}$	$\mathbf{94.77}_{(0.59)}$	$87.94_{(1.69)}$

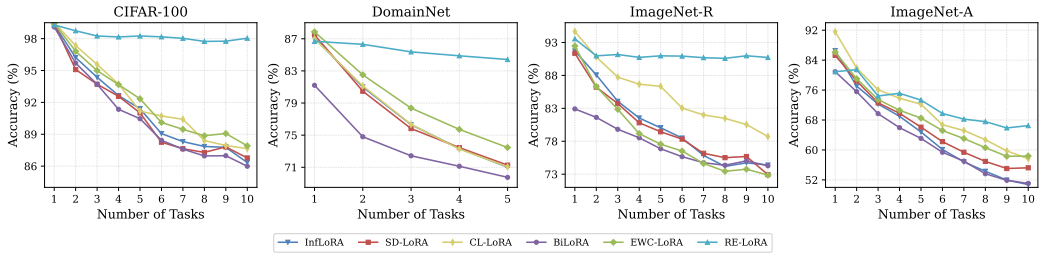


Figure 2: Task-wise performance comparison of different methods across various datasets.

tasks. To further characterize the stability–plasticity trade-off, we additionally report the normalized *stability* and *plasticity* scores of EWC-LoRA (Zheng et al. (2026)): stability quantifies how much of each task's peak accuracy is retained by the end of the sequence, while plasticity quantifies how close the accuracy obtained right after learning a task is to that of a model fine-tuned exclusively on it. Both scores are normalized and thus directly comparable across datasets and methods; their formal definitions are deferred to Appendix A.3. Last, to better contextualize performance, we report both an upper target (i.e., *Full Expansion*) and a lower target (i.e., *Finetune*). The upper target expands vanilla LoRA to all tasks and uses our PGTR to isolate logits from different tasks, while the lower target sequentially trains on tasks without any forgetting mitigation. The implementation details of RE-LoRA and the descriptions of competing methods are provided in Appendix A.

3.1 Experimental Results

Comparison with Various CL Baselines. Tab. 1 shows that RE-LoRA attains the best results among all baseline methods on all four datasets. Prompt-based methods rely on an expanding prompt pool that becomes unreliable as it grows, which is especially harmful on the fine-grained ImageNet-A (HiDe-Prompt 56.32% vs. RE-LoRA 72.32% in Avg.). LoRA-expansion methods keep one component per task and still trail RE-LoRA (CL-LoRA 85.20% vs. RE-LoRA 91.16% in Avg. on ImageNet-R). Shared-LoRA methods keep a constant parameter count, yet their regularization or gradient projection is less effective and forgetting stays severe (EWC-LoRA 73.46% vs. RE-LoRA 84.42% in $\bar{A}_5$ on DomainNet). RE-LoRA thus outperforms all baselines while remaining rehearsal-free, regularization-free, and budget-constant, because it consolidates rather than expands knowledge: the single adapter is reparameterized at a fixed rank by merge-and-compress, and prototype-guided routing eliminates the prediction bias that survives feature-level adaptation. Fig. 2 further traces accuracy on earlier tasks along the sequence: RE-LoRA's curves stay nearly flat and decay the least, whereas the baselines decline steadily.

Stability and Plasticity Analysis. Tab. 2 evaluates the stability and plasticity of the LoRA-based methods. RE-LoRA attains the highest stability on all four datasets and the highest plasticity on three, while the baselines exhibit the expected trade-off: InfLoRA is stability-oriented but collapses in plasticity on the fine-grained ImageNet-A (72.69%), SD-LoRA favors plasticity but shows the lowest stability (82.45% on DomainNet), and EWC-LoRA is balanced yet inferior to RE-LoRA on both axes. On ImageNet-A, RE-LoRA's plasticity (87.94%) is slightly below SD-LoRA (92.13%) and EWC-LoRA (92.78%), reflecting a deliberate prioritization of stability on this hard benchmark. Overall, RE-LoRA is the only method that is simultaneously near-ceiling on both metrics.

Results across Varied Task Lengths. Tab. 3 compares ImageNet-R under 5-task and 20-task splits, highlighting the accumulation of forgetting. RE-LoRA maintains or even improves its performance as the sequence lengthens, whereas nearly all baselines degrade. This contrast exemplifies the stability–plasticity–efficiency dilemma: prompt methods accumulate task-specific parameters whose selection becomes ambiguous as the sequence grows (CODA-Prompt 80.30%→73.95% in

Table 3: Comparison of results on ImageNet-R with different task lengths.

Methods	ImageNet-R 5 tasks		ImageNet-R 20 tasks	
	$\bar{A}_5(\uparrow)$	Avg.$(\uparrow)$	$\bar{A}_{20}(\uparrow)$	Avg.$(\uparrow)$
Full Expansion	$92.05_{(0.25)}$	$91.75_{(0.13)}$	$95.35_{(0.03)}$	$95.49_{(0.09)}$
Finetune	$69.26_{(0.74)}$	$78.88_{(0.31)}$	$47.06_{(2.05)}$	$63.01_{(0.56)}$
L2P	$73.04_{(0.71)}$	$76.94_{(0.41)}$	$68.97_{(0.51)}$	$74.16_{(0.32)}$
DualPrompt	$69.99_{(0.57)}$	$72.24_{(0.41)}$	$65.23_{(0.45)}$	$71.30_{(0.16)}$
CODA-Prompt	$76.63_{(0.27)}$	$80.30_{(0.28)}$	$69.38_{(0.33)}$	$73.95_{(0.63)}$
HiDe-Prompt	$74.77_{(0.25)}$	$78.15_{(0.24)}$	$73.59_{(0.19)}$	$77.93_{(0.19)}$
InfLoRA	$76.95_{(0.23)}$	$81.81_{(0.14)}$	$69.89_{(0.56)}$	$76.68_{(0.57)}$
SD-LoRA	$79.15_{(0.20)}$	$83.01_{(0.42)}$	$75.26_{(0.37)}$	$80.22_{(0.72)}$
LoRA-P&M	$81.47_{(0.56)}$	$85.96_{(0.52)}$	$76.37_{(0.09)}$	$82.77_{(0.71)}$
MoCL	$\underline{82.29}_{(0.11)}$	$\underline{86.32}_{(0.09)}$	$\underline{79.22}_{(0.17)}$	$\underline{85.03}_{(0.20)}$
EWC-LoRA	$76.36_{(0.21)}$	$81.43_{(0.13)}$	$70.18_{(1.06)}$	$77.06_{(0.54)}$
RE-LoRA	$\mathbf{88.57}_{(0.34)}$	$\mathbf{88.81}_{(0.29)}$	$\mathbf{92.00}_{(0.61)}$	$\mathbf{94.71}_{(0.19)}$

Table 4: Comparison of results on different self-supervised backbones.

PTMs	Method	CIFAR-100 10 tasks		ImageNet-R 10 tasks	
		$\bar{A}_{10}(\uparrow)$	Avg.$(\uparrow)$	$\bar{A}_{10}(\uparrow)$	Avg.$(\uparrow)$
DINO	L2P	$70.65_{(0.57)}$	$79.02_{(0.01)}$	$57.40_{(0.23)}$	$62.56_{(0.20)}$
	DualP	$74.90_{(0.21)}$	$83.98_{(0.16)}$	$58.57_{(0.45)}$	$64.89_{(0.15)}$
	CODA-P	$77.50_{(0.64)}$	$84.81_{(0.30)}$	$63.15_{(0.39)}$	$69.73_{(0.25)}$
	HiDe-P	$\underline{92.51}_{(0.11)}$	$\underline{94.25}_{(0.01)}$	$68.11_{(0.18)}$	$71.70_{(0.01)}$
	SD-LoRA	$87.58_{(0.28)}$	$91.78_{(0.19)}$	$\underline{78.06}_{(0.31)}$	$\underline{74.31}_{(0.27)}$
	RE-LoRA	$\mathbf{97.22}_{(0.17)}$	$\mathbf{97.37}_{(0.15)}$	$\mathbf{85.33}_{(0.59)}$	$\mathbf{86.03}_{(0.29)}$
iBOT	L2P	$75.57_{(0.41)}$	$82.69_{(0.06)}$	$60.97_{(0.26)}$	$65.95_{(0.02)}$
	DualP	$76.63_{(0.05)}$	$85.08_{(0.12)}$	$61.51_{(1.05)}$	$67.11_{(0.08)}$
	CODA-P	$79.11_{(1.02)}$	$86.21_{(0.49)}$	$66.56_{(0.68)}$	$73.14_{(0.57)}$
	HiDe-P	$\underline{93.48}_{(0.11)}$	$\underline{95.02}_{(0.01)}$	$71.33_{(0.21)}$	$73.62_{(0.13)}$
	SD-LoRA	$87.95_{(0.24)}$	$92.47_{(0.16)}$	$\underline{78.33}_{(0.28)}$	$\underline{74.12}_{(0.25)}$
	RE-LoRA	$\mathbf{97.37}_{(0.05)}$	$\mathbf{97.58}_{(0.03)}$	$\mathbf{87.42}_{(0.03)}$	$\mathbf{87.92}_{(0.05)}$

Table 5: Ablation analysis of the proposed RE-LoRA. "$*$" denotes the performance with task identifiers provided. "Learnable" refers to the prototypes acquired through the learnable LoRA modules.

Variants	ImageNet-A 10 tasks				ImageNet-R 10 tasks			
	$\bar{A}_{10}(\uparrow)$	Avg.$(\uparrow)$	$\bar{A}^*_{10}(\uparrow)$	Avg.$^*(\uparrow)$	$\bar{A}_{10}(\uparrow)$	Avg.$(\uparrow)$	$\bar{A}^*_{10}(\uparrow)$	Avg.$^*(\uparrow)$
Vanilla LoRA	$26.37_{(0.24)}$	$40.36_{(0.43)}$	$63.63_{(1.54)}$	$68.95_{(0.87)}$	$28.72_{(1.91)}$	$44.11_{(2.15)}$	$83.59_{(3.09)}$	$85.80_{(2.50)}$
+ RAL	$30.45_{(3.82)}$	$42.40_{(0.67)}$	$\mathbf{66.72}_{(1.40)}$	$71.54_{(0.67)}$	$29.72_{(1.76)}$	$50.89_{(0.14)}$	$\mathbf{90.96}_{(0.18)}$	$\mathbf{91.28}_{(0.26)}$
+ HKC	$29.37_{(0.83)}$	$48.22_{(0.18)}$	$66.50_{(1.29)}$	$72.32_{(0.70)}$	$39.26_{(4.58)}$	$57.69_{(0.41)}$	$90.72_{(0.15)}$	$91.16_{(0.03)}$
+ PGTR (Learnable)	$43.15_{(4.55)}$	$63.69_{(2.66)}$	$66.50_{(1.29)}$	$72.32_{(0.70)}$	$59.54_{(4.55)}$	$71.41_{(0.11)}$	$90.72_{(0.15)}$	$91.16_{(0.03)}$
+ PGTR (RE-LoRA)	$\mathbf{66.50}_{(1.29)}$	$\mathbf{72.32}_{(0.70)}$	$66.50_{(1.29)}$	$\mathbf{72.32}_{(0.70)}$	$\mathbf{90.72}_{(0.15)}$	$\mathbf{91.16}_{(0.03)}$	$90.72_{(0.15)}$	$91.16_{(0.03)}$

Avg.), while regularization-based LoRA methods accumulate Fisher-space constraints that progressively restrict plasticity (EWC-LoRA 81.43%→77.06% in Avg.). RE-LoRA avoids both pitfalls: HKC folds every task's residual adaptation into the same adapter and re-compresses the merged adaptation to a fixed rank by the Eckart–Young optimal approximation, so each new task adds neither task-specific parameters nor regularization constraints and the evolving adapter keeps absorbing new knowledge without obvious decay. This is consistent with Proposition 2.2 and Corollary 2.3, which show that the cumulative consolidation error grows at most linearly with the number of tasks, with a per-task constant that decays exponentially with the rank budget r_{base}.

Generalization Performance across Self-supervised Backbones. Tab. 4 replaces the supervised ViT-B/16 with self-supervised DINO (Caron et al. (2021)) and iBOT (Zhou et al. (2021)) backbones. RE-LoRA remains the strongest in every column, with gains especially pronounced on ImageNet-R. This insensitivity to the pre-training objective follows from our design: RAL and HKC act on the geometry of the low-rank updates rather than on backbone-specific supervised cues, and PGTR relies on frozen-backbone prototypes whose feature statistics stay stable regardless of the self-supervised recipe. The framework therefore transfers to DINO and iBOT without modification.

Ablation Studies. Tab. 5 isolates each component on ImageNet-A and ImageNet-R under two settings: Class-IL, in which no task identifier is available at inference, and Task-IL, in which task identifiers are provided and the metrics are marked by "$*$". Vanilla sequential LoRA suffers from both catastrophic forgetting and prediction bias under Class-IL, and also underperforms on Task-IL because each new task sequentially overwrites previous knowledge. Adding RAL entails per-task expansion but improves both regimes by additively composing each task's update with the frozen historical adapter: the Class-IL Avg. rises from 40.36% and 44.11% to 42.40% and 50.89%, and the Task-IL Avg. from 68.95% and 85.80% to 71.54% and 91.28%, confirming that residual composition preserves old knowledge. Adding HKC keeps the parameter count constant while further raising the Class-IL Avg. to 48.22% and 57.69% and maintaining or even enhancing Task-IL Avg. (72.32% and 91.16%), showing that HKC preserves consolidated knowledge and improves parameter efficiency. For PGTR, we compared LoRA-extracted prototypes against frozen-backbone prototypes for routing; the latter substantially outperformed the former. Because RE-LoRA maintains a single evolving LoRA, its features for later tasks inevitably overlap with those of historical tasks, which interferes with routing. Across all evaluated datasets, frozen-prototype routing reaches 100% accuracy, matching the Task-IL setting. In short, RAL and HKC improve the learned representation, mitigate forgetting, and enhance the parameter efficiency, while PGTR eliminates prediction bias.

Table 6: Comparison on ImageNet-R (20 tasks) of parameter, storage, and inference efficiency. All ratios are normalized to a single LoRA module, where 1.00× denotes no growth or additional overhead. Stored features include class prototypes and curvature/Fisher statistics.

Methods	Avg.(↑)	Additional Parameters(× single) (↓)	Stored Features (M)(↓)	Additional Inference Latency (× single) (↓)
InfLoRA	76.68	1.00×	0.10	1.00×
SD-LoRA	80.22	20.00×	0.00	1.00×
LoRA-P&M	82.77	20.00×	7.40	1.00×
CL-LoRA	84.77	10.50×	0.15	10.50×
MoCL	85.03	1.00×	8.23	1.00×
EWC-LoRA	77.06	1.00×	14.16	1.00×
RE-LoRA	94.71	1.00×	0.15	∼2.00×

Analysis of Computation, Parameter, and Storage Efficiency. Tab. 6 compares the parameter, storage, and inference efficiency of the LoRA-based methods at the inference stage on ImageNet-R's 20^{th} task. SD-LoRA and LoRA-P&M retain one adapter per task and grow to $20.00\times$ additional parameters. Though CL-LoRA shares one adapter in the first six ViT layers, stores each task's specific adapter in the remaining six still reaches $10.50\times$. In contrast, InfLoRA, EWC-LoRA, MoCL, and RE-LoRA keep a constant parameter footprint of $1.00\times$. For stored features, SD-LoRA keeps none, CL-LoRA and RE-LoRA store only class prototypes (0.15M), and the remaining methods trade storage overhead for knowledge preservation: LoRA-P&M stores a per-task diagonal Fisher (7.40M), MoCL a consolidated curvature and momentum subspace (8.23M), and EWC-LoRA a full-dimensional accumulated Fisher (14.16M), all of which add considerable storage overhead. At inference, only CL-LoRA incurs substantial overhead ($10.50\times$) because it forwards through every retained task-specific adapter, whereas RE-LoRA adds a single frozen-backbone pass for prototype routing ($\sim 2.00\times$). RE-LoRA therefore achieves the best performance, constant parameter budget, and low storage of the most efficient baselines, storing only prototypes and one evolving adapter.

4 Related Works

Baselines. Prompt-based CIL methods adapt a frozen ViT with learnable tokens: L2P (Wang et al. (2022b)) selects instance-specific prompts by key matching, DualPrompt (Wang et al. (2022a)) separates task-invariant from task-specific prompts, CODA-Prompt (Smith et al. (2023)) replaces discrete selection with end-to-end attention, and HiDe-Prompt (Wang et al. (2023)) decomposes the CIL objective into within-task prediction, task-identity inference, and task-adaptive prediction. Their prompt pool expands with the task sequence, however, which increases storage and makes prompt retrieval increasingly ambiguous and costly at inference (Wu et al. (2025); Lai et al.). **LoRA-based CIL methods** either allocate task-specific low-rank components (e.g., SD-LoRA (Wu et al. (2025)), CL-LoRA (He et al. (2025)), and BiLoRA (Zhu et al. (2025))) whose parameter footprint grows with the sequence, or keep a shared adapter regularized by importance statistics (e.g., InfLoRA (Liang & Li (2024)), EWC-LoRA (Zheng et al. (2026)) and MoCL (Lai et al.)), which adds computation and memory and progressively restricts plasticity as constraints accumulate. Model-merging approaches such as LoRA-P&M (Qiu et al. (2025)) combine trained weights by scalar interpolation or component accumulation without enforcing a fixed rank budget.

Ours. RE-LoRA departs from all baselines: it evolves a single constant-size adapter through additive residual learning and optimal low-rank consolidation, and routes tasks with frozen-backbone prototypes, avoiding both parameter expansion and auxiliary regularization. A detailed discussion is provided in Appendix D.

5 Conclusion

We presented Residual Evolving LoRA (RE-LoRA), a rehearsal-free class-incremental learning framework that resolves the stability–plasticity–efficiency dilemma with a single evolving low-rank adapter of constant size. Residual Adapter Learning composes each task's update additively with the frozen historical adapter, so new tasks refine rather than overwrite prior knowledge without auxiliary regularization; Historical Knowledge Consolidation merges the residual into the historical adapter and re-compresses it by truncated SVD, which we prove is the optimal rank-r_{base} approximation of the combined adaptation and whose truncation error provably accumulates at most linearly rather than compounding over the task sequence; and Prototype-Guided Task Routing removes prediction bias at inference using stable frozen-backbone prototypes. Extensive experiments demonstrate that RE-LoRA consistently outperforms state-of-the-art LoRA-based CIL baselines while keeping the parameter footprints constant.

AI USE STATEMENT

Generative AI tools were used for language editing. All technical claims, implementations, experiments, and reported numbers were verified by the authors.

REFERENCES

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, pp. 177–186. Springer, 2010.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9630–9640. IEEE, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Jiangpeng He, Zhihao Duan, and Fengqing Zhu. Cl-lora: Continual low-rank adaptation for rehearsal-free class-incremental learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30534–30544. IEEE, 2025.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021a.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021b.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Jiajun Lai, Qi Liu, Shijie Li, and Huaiguang Jiang. Mocl: Metabolic optimization for curvature-aware continual learning. In *Forty-third International Conference on Machine Learning*.

Yan-Shuo Liang and Wu-Jun Li. 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.

Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1406–1415, 2019.

Haomiao Qiu, Miao Zhang, Ziyue Qiao, and Liqiang Nie. Train with perturbation, infer after merging: A two-stage framework for continual learning. *arXiv preprint arXiv:2505.22389*, 2025.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11909–11919, 2023.

Gido M van de Ven and Andreas S Tolias. Three continual learning scenarios. In *NeurIPS Continual Learning Workshop*, volume 1, 2018.

Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingyi Huang, Hang Su, and Jun Zhu. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. *Advances in Neural Information Processing Systems*, 36:69054–69076, 2023.

Yan Wang, Da-Wei Zhou, and Han-Jia Ye. Integrating task-specific and universal adapters for pre-trained model-based class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 806–816, 2025.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European conference on computer vision*, pp. 631–648. Springer, 2022a.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 139–149, 2022b.

Hermann Weyl. Inequalities between the two kinds of eigenvalues of a linear transformation. *Proceedings of the national academy of sciences*, 35(7):408–411, 1949.

Yichen Wu, Hongming Piao, Long-Kai Huang, Renzhen Wang, Wanhua Li, Hanspeter Pfister, Deyu Meng, Kede Ma, and Ying Wei. Sd-lora: Scalable decoupled low-rank adaptation for class incremental learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19148–19158, 2023.

Yaoyue Zheng, Yin Zhang, Joost van de Weijer, Gido van de Ven, Shaoyi Du, Xuetao Zhang, and Zhiqiang Tian. Revisiting weight regularization for low-rank continual learning. In *International Conference on Learning Representations*, volume 2026, pp. 42895–42919, 2026.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

Hao Zhu, Yifei Zhang, Junhao Dong, and Piotr Koniusz. Bilora: Almost-orthogonal parameter spaces for continual learning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25613–25622. IEEE, 2025.

Algorithm 1 Training Procedure for **RE-LoRA**

Require: Task stream $\{\mathcal{T}_t\}_{t=1}^{N}$ with data $\{\mathcal{D}_t\}_{t=1}^{N}$, frozen backbone F_Ω, classifier F_ω, fix rank budget r_{base}, training epochs $E_{\max}$, learning rate η

Ensure: Trained LoRA weight $\{\Delta\mathbf{W}_{\text{hist}}^{(\ell)}\}_{\ell=1}^{L}$, trained classifier F_ω, prototypes $\mathbf{P}$

Initialize historical weight $\Delta\mathbf{W}_{\text{hist}}^{(\ell)} \leftarrow \mathbf{0}$; $\mathbf{P} \leftarrow \emptyset$
for $t = 1$ **to** N **do**
  {**Residual Adapter Learning (Sec. 2.2)**}
  Initialize current residual factors $\{\mathbf{A}_{\text{res}}^{(\ell)}\}_{\ell=1}^{L}, \{\mathbf{B}_{\text{res}}^{(\ell)}\}_{\ell=1}^{L}$
  for $e = 1$ **to** $E_{\max}$ **do**
    Assemble effective update $\{\Delta\mathbf{W}_{\text{eff}}^{(\ell)}\}_{\ell=1}^{L}$ via Eq. (1)
    Forward feature $\{\mathbf{f}^{(\ell)}\}_{\ell=1}^{L}$ via Eq. (2)
    Make prediction through F_ω
    Update $\{\mathbf{A}_{\text{res}}^{(\ell)}\}_{\ell=1}^{L}$, $\{\mathbf{B}_{\text{res}}^{(\ell)}\}_{\ell=1}^{L}$ and F_ω by $\eta\nabla\mathcal{L}_{\text{CE}}$
  end for
  {**Historical Knowledge Consolidation (Sec. 2.3)**}
  if $t = 1$ **then**
    Freeze and consolidate the trained $\{\Delta\mathbf{W}_{\text{res}}^{(\ell)}\}_{\ell=1}^{L}$ as $\{\Delta\mathbf{W}_{\text{hist}}^{(\ell)}\}_{\ell=1}^{L}$
  else
    Merge and compress $\{\Delta\mathbf{W}_{\text{hist}}^{(\ell)} + \Delta\mathbf{W}_{\text{res}}^{(\ell)}\}_{\ell=1}^{L}$ via Eq. (3) to update $\{\Delta\mathbf{W}_{\text{hist}}^{(\ell)}\}_{\ell=1}^{L}$
  end if
  {**Prototype Construction (Sec. 2.4)**}
  for each class c introduced in $\mathcal{T}_t$ **do**
    Compute $\mathbf{p}_c$ via Eq. (4); $\mathbf{P} \leftarrow \mathbf{P} \cup \{\mathbf{p}_c\}$
  end for
end for
return $\{\Delta\mathbf{W}_{\text{hist}}^{(\ell)}\}_{\ell=1}^{L}$, F_ω, $\mathbf{P}$

A Additional Details of RE-LoRA

A.1 Algorithmic Summary

Algorithm 1 summarizes the complete pipeline of the proposed RE-LoRA. Given a task stream and a frozen backbone, RE-LoRA initializes the historical adapter to zero and the prototype matrix to empty as the initialization for task $\mathcal{T}_1$. For each subsequent task $\mathcal{T}_t$, RE-LoRA first instantiates a fresh rank-r_{base} residual adapter $\Delta\mathbf{W}_{\text{res}} = \mathbf{B}_{\text{res}}\mathbf{A}_{\text{res}}$ and, for $E_{\max}$ epochs, assembles the effective update $\Delta\mathbf{W}_{\text{eff}} = \Delta\mathbf{W}_{\text{hist}} + \mathbf{B}_{\text{res}}\mathbf{A}_{\text{res}}$ (the proposed RAL, Eq. (1)), performs a forward pass and prediction, and updates only the residual factors $\mathbf{A}_{\text{res}}, \mathbf{B}_{\text{res}}$ and classifier F_ω via $\eta\nabla\mathcal{L}_{\text{CE}}$; the historical adapter $\Delta\mathbf{W}_{\text{hist}}$ remains frozen, so no replay buffer or auxiliary regularization is required. After training, the proposed HKC performs knowledge consolidation: for $\mathcal{T}_1$, the learned residual adapters are frozen directly as the historical adapters of $\mathcal{T}_2$, while for every subsequent task, the historical and residual adapters are merged and re-compressed to rank r_{base} by truncated SVD (Eq. (3)), keeping the adapter size constant. Finally, after training of each task, the prototype of each newly introduced class is computed from frozen-backbone features and appended to $\mathbf{P}$. The procedure returns the single evolved adapter, the shared classifier, and the prototypes used for the proposed PGTR.

A.2 Implementation Details of RE-LoRA

We adopt ViT-B/16 (Dosovitskiy et al. (2020)) pre-trained on ImageNet-21K then fine-tuned on ImageNet-1K as the initialization for backbone. We also experiment with a self-supervised ViT-B/16 from DINO (Caron et al. (2021)) and iBOT (Zhou et al. (2021)). LoRA modules (r_{base}=10) are inserted into the query and value projections of all multi-head attention layers, identical to the other LoRA baselines. Training uses SGD optimizer (Bottou (2010)) with a learning rate of 0.03, a batch size of 32, and runs for 10 epochs per task. Results report the mean and standard deviation over five runs on a single NVIDIA RTX 3090 GPU.

A.3 Stability and Plasticity Metrics

To characterize the stability–plasticity trade-off beyond average accuracy, we adopt the normalized stability and plasticity scores introduced in EWC-LoRA (Zheng et al. (2026)). The *stability* score is the relative drop of each task's accuracy from its peak to the end of the sequence:

$$\text{Stability} = 1 - \mathcal{F} = 1 - \frac{1}{N-1}\sum_{i=1}^{N-1}\frac{\max_{t<N} A_{t,i} - A_{N,i}}{\max_{t<N} A_{t,i}}. \tag{11}$$

The *plasticity* score is the ratio between the accuracy on a task right after learning it (i.e., $A_{i,i}$) and the reference accuracy of a model fine-tuned exclusively on that task (i.e., $A^{\text{ref}}_{i,i}$), averaged over all tasks:

$$\text{Plasticity} = \frac{1}{N}\sum_{i=1}^{N}\frac{A_{i,i}}{A^{\text{ref}}_{i,i}}. \tag{12}$$

Both scores are normalized and thus independent of the absolute performance on a dataset, enabling direct comparison across datasets and methods.

B Proofs and Empirical Validation of Proposition 2.2 and Corollary 2.3

B.1 Proof of Proposition 2.2

We first prove the identity in Eq. (8) by induction on t. For $t = 1$, $\mathbf{H}_0 = \mathbf{0}$, and thus

$$\mathbf{F}_1 - \mathbf{H}_1 = \mathbf{R}_1 - T_{r_{\text{base}}}(\mathbf{R}_1) = (\mathbf{H}_0 + \mathbf{R}_1) - T_{r_{\text{base}}}(\mathbf{H}_0 + \mathbf{R}_1) = \mathbf{E}_1.$$

Assume that $\mathbf{F}_{t-1} - \mathbf{H}_{t-1} = \sum_{s=1}^{t-1}\mathbf{E}_s$. Then,

$$\begin{aligned}\mathbf{F}_t - \mathbf{H}_t &= (\mathbf{F}_{t-1} + \mathbf{R}_t) - T_{r_{\text{base}}}(\mathbf{H}_{t-1} + \mathbf{R}_t)\\ &= (\mathbf{F}_{t-1} - \mathbf{H}_{t-1}) + [(\mathbf{H}_{t-1} + \mathbf{R}_t) - T_{r_{\text{base}}}(\mathbf{H}_{t-1} + \mathbf{R}_t)]\\ &= \sum_{s=1}^{t-1}\mathbf{E}_s + \mathbf{E}_t = \sum_{s=1}^{t}\mathbf{E}_s,\end{aligned} \tag{13}$$

which completes the induction and establishes Eq. (8).

For the error bound, let $\mathbf{M}_t = \mathbf{H}_{t-1} + \mathbf{R}_t$. Since both $\mathbf{H}_{t-1}$ and $\mathbf{R}_t$ are rank-r_{base} matrices, we have

$$\text{rank}(\mathbf{M}_t) \le \text{rank}(\mathbf{H}_{t-1}) + \text{rank}(\mathbf{R}_t) \le 2r_{\text{base}}.$$

The rank-r_{base} truncation $T_{r_{\text{base}}}(\mathbf{M}_t)$ is the optimal rank-r_{base} approximation of $\mathbf{M}_t$ under the Frobenius norm according to the Eckart–Young theorem. Therefore, the truncation error

$$\mathbf{E}_t = \mathbf{M}_t - T_{r_{\text{base}}}(\mathbf{M}_t)$$

contains only the singular components corresponding to $\sigma_{r_{\text{base}}+1}(\mathbf{M}_t), \ldots, \sigma_{2r_{\text{base}}}(\mathbf{M}_t)$. Consequently,

$$\|\mathbf{E}_t\|_F^2 = \sum_{i=r_{\text{base}}+1}^{2r_{\text{base}}}\sigma_i^2(\mathbf{M}_t) \le r_{\text{base}}\sigma^2_{r_{\text{base}}+1}(\mathbf{M}_t),$$

which gives

$$\|\mathbf{E}_t\|_F \le \sqrt{r_{\text{base}}}\,\sigma_{r_{\text{base}}+1}(\mathbf{M}_t).$$

It remains to bound the $(r_{\text{base}}+1)$-th singular value of $\mathbf{M}_t$. Since $\mathbf{H}_{t-1}$ has rank at most r_{base}, we have

$$\sigma_{r_{\text{base}}+1}(\mathbf{H}_{t-1}) = 0.$$

Applying Weyl's inequality (Weyl (1949)) for singular values to $\mathbf{M}_t = \mathbf{H}_{t-1} + \mathbf{R}_t$ yields

$$\sigma_{r_{\text{base}}+1}(\mathbf{M}_t) \le \sigma_{r_{\text{base}}+1}(\mathbf{H}_{t-1}) + \sigma_1(\mathbf{R}_t) = \sigma_1(\mathbf{R}_t).$$

Because the spectral norm is bounded by the Frobenius norm,

$$\sigma_1(\mathbf{R}_t) = \|\mathbf{R}_t\|_2 \le \|\mathbf{R}_t\|_F,$$

we obtain

$$\|\mathbf{E}_t\|_F \le \sqrt{r_{\text{base}}}\|\mathbf{R}_t\|_F.$$

Finally, applying the triangle inequality to Eq. (8), we have

$$\|\mathbf{F}_N - \mathbf{H}_N\|_F = \left\|\sum_{t=1}^{N} \mathbf{E}_t\right\|_F \le \sum_{t=1}^{N} \|\mathbf{E}_t\|_F \le \sqrt{r_{\text{base}}} \sum_{t=1}^{N} \|\mathbf{R}_t\|_F,$$

which completes the proof.

B.2 Proof of Corollary 2.3

Let $\mathbf{M}_t = \mathbf{H}_{t-1} + \mathbf{R}_t$. Because both $\mathbf{H}_{t-1}$ and $\mathbf{R}_t$ are rank-r_{base} matrices, we have

$$\text{rank}(\mathbf{M}_t) \le \text{rank}(\mathbf{H}_{t-1}) + \text{rank}(\mathbf{R}_t) \le 2r_{\text{base}}.$$

Therefore, all singular values of $\mathbf{M}_t$ after the first $2r_{\text{base}}$ components are zero. Since $\mathbf{E}_t = \mathbf{M}_t - T_{r_{\text{base}}}(\mathbf{M}_t)$ corresponds to the discarded singular components after rank-r_{base} truncation, the Eckart–Young theorem gives

$$\|\mathbf{E}_t\|_F^2 = \sum_{i=r_{\text{base}}+1}^{2r_{\text{base}}} \sigma_i^2(\mathbf{M}_t).$$

Under the spectral decay assumption

$$\sigma_i(\mathbf{M}_t) \le \alpha\rho^{i-1},$$

we obtain

$$\begin{aligned}\|\mathbf{E}_t\|_F^2 &= \sum_{i=r_{\text{base}}+1}^{2r_{\text{base}}} \sigma_i^2(\mathbf{M}_t) \le \sum_{i=r_{\text{base}}+1}^{2r_{\text{base}}} \alpha^2\rho^{2(i-1)} \le \sum_{i=r_{\text{base}}+1}^{\infty} \alpha^2\rho^{2(i-1)} \\ &= \alpha^2\rho^{2r_{\text{base}}} \sum_{k=0}^{\infty} \rho^{2k} = \frac{\alpha^2\rho^{2r_{\text{base}}}}{1-\rho^2}.\end{aligned}$$

Taking square roots yields

$$\|\mathbf{E}_t\|_F \le \frac{\alpha\rho^{r_{\text{base}}}}{\sqrt{1-\rho^2}}.$$

Finally, applying the non-compounding error bound in Eq. (9), we have

$$\|\mathbf{F}_N - \mathbf{H}_N\|_F \le \sum_{t=1}^{N} \|\mathbf{E}_t\|_F \le \sum_{t=1}^{N} \frac{\alpha\rho^{r_{\text{base}}}}{\sqrt{1-\rho^2}} = \frac{\alpha\rho^{r_{\text{base}}}}{\sqrt{1-\rho^2}} N,$$

which completes the proof.

B.3 Empirical Validation on Error Accumulation and Spectral Decay

The two results above make two testable statements: (i) the consolidation error does not compound, i.e., $\|\mathbf{F}_N - \mathbf{H}_N\|_F = \|\sum_t \mathbf{E}_t\|_F$, so that its growth is governed by the per-step truncation residuals; and (ii) under the geometric spectral decay $\sigma_i(\mathbf{M}_t) \le \alpha\rho^{i-1}$ of the pre-truncation merged matrices, the cumulative error grows at most linearly in N. We now verify both statements empirically on CIFAR-100, ImageNet-A, and ImageNet-R (10 tasks each), using the same ViT-B/16 protocol as the main experiments.

Experimental design. We instrument the HKC step inside the backbone. Whenever a task finishes, we hold the trained residual $\mathbf{R}_t$ and the inherited historical adapter $\mathbf{H}_{t-1}$; before truncation we record the dense merged matrix $\mathbf{M}_t = \mathbf{H}_{t-1} + \mathbf{R}_t$ together with its singular values $\sigma_i(\mathbf{M}_t)$ for every adapted module. In parallel, we maintain the untruncated reference $\mathbf{F}_t = \mathbf{F}_{t-1} + \mathbf{R}_t$ and

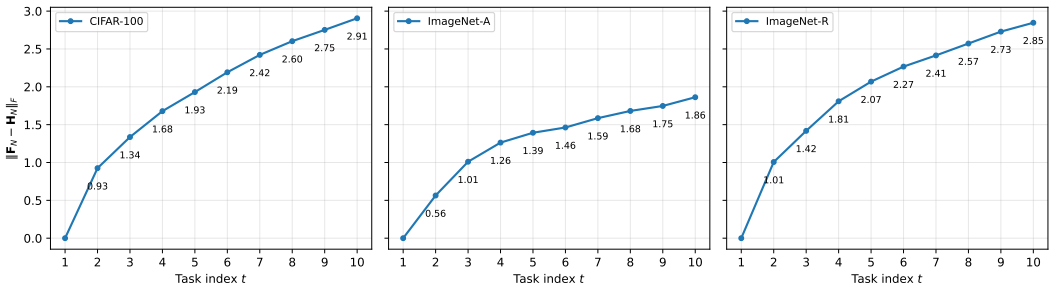


Figure 3: Cumulative consolidation error $\|\mathbf{F}_N - \mathbf{H}_N\|_F$ as a function of the task index on CIFAR-100, ImageNet-A, and ImageNet-R (10 tasks each). The error grows monotonically but sub-linearly, and the per-step increments broadly decrease, as predicted by Proposition 2.2.

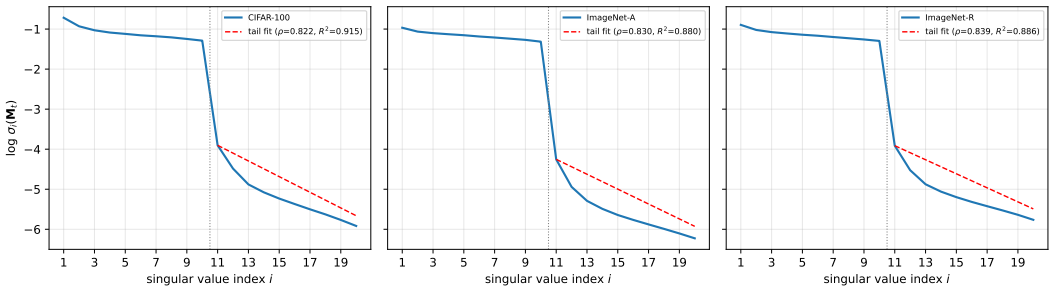


Figure 4: Pooled $\log \sigma_i(\mathbf{M}_t)$ of the pre-truncation merged matrices $\mathbf{M}_t = \mathbf{H}_{t-1} + \mathbf{R}_t$ on CIFAR-100, ImageNet-A, and ImageNet-R. The vertical dotted line marks the rank budget $r_{\text{base}} = 10$, and the red dashed line is the fitted geometric tail $\log \sigma_i \approx \log \alpha + (i-1) \log \rho$ over $i = 11, \ldots, 20$. The discarded tail is approximately geometric, supporting the envelope assumption of Corollary 2.3.

the consolidated adapter $\mathbf{H}_t = T_{r_{\text{base}}}(\mathbf{M}_t)$, and after each task we evaluate the global Frobenius deviation $\|\mathbf{F}_N - \mathbf{H}_N\|_F$ by summing the squared per-module differences over all layers and taking the square root. As in the main experiments, LoRA is applied to the query and value projections of all 12 blocks with rank budget $r_{\text{base}} = 10$, so every $\mathbf{M}_t$ has rank at most $2r_{\text{base}} = 20$.

Spectral analysis protocol. The geometric-decay assumption of Corollary 2.3 states that $\sigma_i(\mathbf{M}_t) \leq \alpha \rho^{i-1}$. Reading it as an approximate equality and taking logarithms of both sides gives

$$\log \sigma_i(\mathbf{M}_t) \approx \log \alpha + (i-1) \log \rho, \tag{14}$$

since $\log(\alpha \rho^{i-1}) = \log \alpha + (i-1) \log \rho$. Therefore, if the decay is geometric, the logarithm of the spectrum is an affine function of the index i whose intercept is $\log \alpha$ and whose slope is $\log \rho$, so the slope directly recovers the decay ratio as $\rho = e^{\text{slope}}$. We hence pool $\log \sigma_i(\mathbf{M}_t)$ over all tasks and modules, plot it against i, and fit the line in Eq. (14) by ordinary least squares on the tail $i = r_{\text{base}} + 1, \ldots, 2r_{\text{base}}$, which contains exactly the components discarded by the truncation and thus determines $\|\mathbf{E}_t\|_F$. To quantify how well the geometric model describes the tail, we report the coefficient of determination

$$R^2 = 1 - \frac{\text{SS}_{\text{res}}}{\text{SS}_{\text{tot}}}, \qquad \text{SS}_{\text{res}} = \sum_j (y_j - \hat{y}_j)^2, \qquad \text{SS}_{\text{tot}} = \sum_j (y_j - \bar{y})^2, \tag{15}$$

where $y_j = \log \sigma_{i_j}(\mathbf{M}_t)$ are the observed log-singular values, $\hat{y}_j$ are the values predicted by the fitted line, and $\bar{y}$ is their mean. Because the fitted model contains an intercept, R^2 lies in $[0, 1]$ and equals the squared Pearson correlation between i and $\log \sigma_i$; it is the fraction of the variance of $\log \sigma_i$ explained by the geometric model, so $R^2 \to 1$ indicates a tail that is almost exactly geometric, whereas smaller values indicate deviations from it. We finally compute the smallest envelope constant α for which $\sigma_i(\mathbf{M}_t) \leq \alpha \rho^{i-1}$ holds for all i, t, and modules.

Error accumulation. Fig. 3 plots $\|\mathbf{F}_N - \mathbf{H}_N\|_F$ against the task index. The measured error increases monotonically and is visibly concave, ending at 2.91 (CIFAR-100), 1.86 (ImageNet-A), and 2.85 (ImageNet-R) after ten tasks. The per-step truncation residuals $\|\mathbf{E}_t\|_F$ follow a broadly decreasing trend up to small fluctuations, from 0.93 to 0.15 (CIFAR-100), 0.56 to 0.12 (ImageNet-A), and 1.01 to 0.12 (ImageNet-R). No obvious amplification across tasks is observed: the contribution

of each consolidation is bounded by its own residual, exactly as Proposition 2.2 prescribes. Because the increments decay rather than grow, the cumulative curve is not merely at most linear but sub-linear over the sequence.

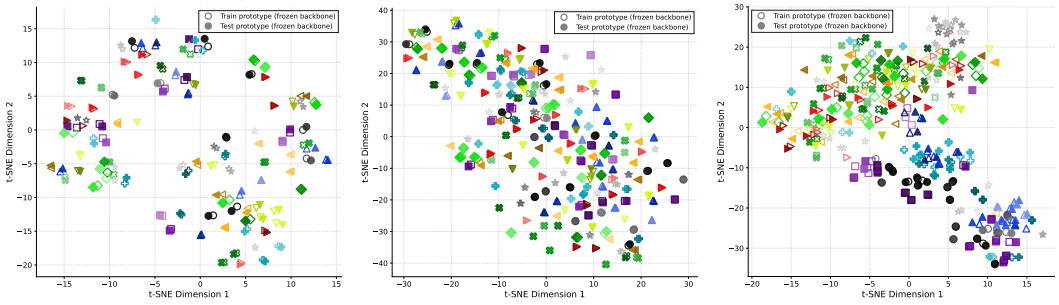
Spectral decay. Fig. 4 shows the pooled $\log \sigma_i(\mathbf{M}_t)$ for the three datasets. The spectrum is clearly two-block: the first $r_{\text{base}} = 10$ components form a slowly decaying block that dominates the energy and is retained by the truncation, followed by a sharp drop and a rapidly decaying tail for $i = 11, \ldots, 20$. This tail is well described by a geometric law, with $\rho = 0.822$ ($R^2 = 0.915$) on CIFAR-100, $\rho = 0.830$ ($R^2 = 0.880$) on ImageNet-A, and $\rho = 0.839$ ($R^2 = 0.886$) on ImageNet-R. The geometric model is therefore best interpreted as an upper envelope on the discarded tail, which is precisely the sense in which the assumption of Corollary 2.3 is used.

Theory verification. Combining the fitted tail ratio with the tightest envelope constant ($\alpha \approx 2.22$, 1.76, and 1.66 for CIFAR-100, ImageNet-A, and ImageNet-R), Corollary 2.3 predicts the per-task constant $C = \alpha \rho^{r_{\text{base}}} / \sqrt{1 - \rho^2} \approx 0.55$, 0.49, and 0.53, hence a linear envelope $C \cdot N \approx 5.54$, 4.91, and 5.27 at $N = 10$. The measured errors 2.91, 1.86, and 2.85 all lie below these bounds (by factors $1.9\times$, $2.6\times$, and $1.8\times$), so the empirical behavior is consistent with the predicted at-most-linear growth. The bound is deliberately loose. The leading singular values are nearly flat, so α is set by them rather than by the much smaller discarded tail, and ρ is close to 1, which inflates $1/\sqrt{1 - \rho^2}$. Geometric decay therefore holds only as a loose upper envelope; splitting the spectrum into a retained subspace plus a fast-decaying tail would give much tighter constants. Even so, the two experiments confirm that repeated low-rank consolidation yields a controlled, non-compounding error growing at most linearly with the number of tasks, matching Proposition 2.2 and Corollary 2.3.

On the choice of the reference trajectory. The reference $\mathbf{F}_N$ above should not be conflated with an independently trained model that never applies truncation. We deliberately define $\mathbf{F}_N$ as the untruncated accumulation of the residuals that are actually generated along the training trajectory, which isolates the error contributed by the consolidation operator alone while keeping the learning trajectory fixed. Let $\mathbf{G}_t = \mathbf{G}_{t-1} + \mathbf{R}'_t$ denote the adapter of an independently trained model that never applies truncation, where a fresh residual $\mathbf{R}'_t$ is trained against the uncompressed history $\mathbf{G}_{t-1}$ at every step. Measuring the discrepancy between RE-LoRA and $\mathbf{G}_N$ by a weight-space Frobenius distance is not a well-posed comparison, for two reasons. First, the two models live in parameter spaces of different dimensions. The consolidated adapter $\mathbf{H}_N$ has rank at most r_{base}, whereas $\mathbf{G}_N$ attains rank up to $N r_{\text{base}}$, so a Frobenius distance between matrices of different rank is not scale-comparable. Second, a higher-rank model is intrinsically more expressive, and therefore a large weight-space distance to it is expected; if anything, it indicates that the compression is effective rather than that RE-LoRA is deficient. The theory therefore controls the quantity $\|\mathbf{F}_N - \mathbf{H}_N\|_F$, which is defined against the untruncated accumulation $\mathbf{F}_N = \sum_{t=1}^{N} \mathbf{R}_t$ of the *same* residual sequence. This reference keeps the learning trajectory fixed and isolates the error contributed by the consolidation operator, whereas replacing it with an independently trained model would conflate truncation error with trajectory divergence. The residuals $\mathbf{R}_t$ are themselves functions of the truncated history $\mathbf{H}_{t-1}$ against which they were trained, and the difference between the two trajectories is not controlled by Proposition 2.2.

C Prototype Geometry of Task Routing after Training

Routing is performed in a task-agnostic reference frame. PGTR decides the task of a test sample using the prototypes stored while the task stream is trained. For a class c introduced in task $\mathcal{T}_t$, the stored prototype $\mathbf{p}_c$ of Eq. (4) is the empirical mean of the features of the *training* samples of that class, and these features are extracted by the strictly frozen pre-trained backbone F_Ω rather than by the adapted backbone. Since F_Ω never changes, the prototype set $\mathbf{P}$ that remains after the N-th task has been trained lives in exactly the same coordinate system as at the moment each individual prototype was created: it is independent of the current LoRA adapter, of the task order, and of the number of tasks already consolidated. This differs from the task-identification mechanism in CL-LoRA (He et al. (2025)), where the features extracted by the shared LoRA drift as the model is updated. As demonstrated by the "+PGTR (Learnable)" experiment in Tab. 5, this mechanism performs worse than our method which utilizes frozen class prototypes.



(a) CIFAR-100 (b) ImageNet-R (c) ImageNet-A

Figure 5: t-SNE embedding of the stored training prototypes $\mathbf{p}_c$ of Eq. (4) (hollow) and the test prototypes $\mathbf{q}_c$ of Eq. (16) (filled); color encodes the task with shades for classes within a task, and marker shape distinguishes tasks. Within each task the filled test prototype of a class lies close to its hollow training prototype, while different tasks mostly form separate clusters (some semantically related tasks interleave). This shows the frozen backbone embeds same-task train/test prototypes coherently and keeps tasks apart, the local geometry that nearest-prototype routing relies on. The test prototypes here are a diagnostic device and are never used for routing.

The test prototypes of a task are closest to the stored prototypes of the same task. To probe the geometry of the reference frame, we build a second set of prototypes from the test split dataset with the same frozen backbone. These *test prototypes* are a diagnostic device only: they are never used by the router and never affect a prediction. Let $\mathcal{D}_k^{\mathrm{tr}}$ and $\mathcal{D}_k^{\mathrm{te}}$ be the training and test sets of task k, with class set $\mathcal{C}_k$. For each class c, we form the training prototype $\mathbf{p}_c$ and the test prototype $\mathbf{q}_c$ as the empirical means of the frozen-backbone features of the samples of that class in the two splits,

$$\mathbf{p}_c = \frac{1}{|\mathcal{D}_c^{\mathrm{tr}}|} \sum_{\mathbf{x} \in \mathcal{D}_c^{\mathrm{tr}}} F_\Omega(\mathbf{x}), \qquad \mathbf{q}_c = \frac{1}{|\mathcal{D}_c^{\mathrm{te}}|} \sum_{\mathbf{x} \in \mathcal{D}_c^{\mathrm{te}}} F_\Omega(\mathbf{x}), \tag{16}$$

where $\mathcal{D}_c^{\mathrm{tr}}$ and $\mathcal{D}_c^{\mathrm{te}}$ collect the samples of class c in the two splits. Here $\mathbf{p}_c$ coincides with the stored prototype of Eq. (4), whereas $\mathbf{q}_c$ is a *class mean of test features*, not an individual test feature. The distinction is important: at inference PGTR compares a single test feature $\mathbf{f}^{\mathrm{te}}$ against the stored $\mathbf{p}_c$ as in Eq. (5), while $\mathbf{q}_c$ only summarizes how the test split of a class is placed in the same frozen frame, which is what the following analysis examines.

We summarize the agreement between the two prototype sets in the matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$. For each stored prototype of task k we take its best cosine match among the test prototypes of task k', and average these maxima over the classes of task k:

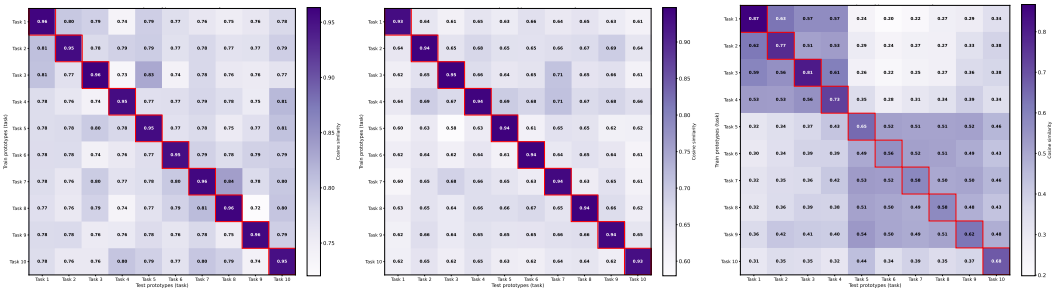
$$\mathbf{S}_{k,k'} = \frac{1}{|\mathcal{C}_k|} \sum_{c \in \mathcal{C}_k} \max_{c' \in \mathcal{C}_{k'}} \frac{\mathbf{p}_c^\top \mathbf{q}_{c'}}{\|\mathbf{p}_c\|_2 \, \|\mathbf{q}_{c'}\|_2}. \tag{17}$$

Because F_Ω is a large-scale pre-trained ViT, features of semantically different tasks are likely to occupy well-separated cones of the embedding space, whereas the training and test splits of one task are two i.i.d. draws from the same semantic distribution and therefore concentrate in a common cone. The measured matrix is supposed to be diagonally dominant,

$$\mathbf{S}_{k,k} > \max_{k' \neq k} \mathbf{S}_{k,k'}, \ \forall\, k \in \{1, \ldots, N\}, \ \text{with margin } \delta = \min_k \left(\mathbf{S}_{k,k} - \max_{k' \neq k} \mathbf{S}_{k,k'} \right) > 0. \tag{18}$$

In short, for every task the average best-match similarity to its own stored prototypes is supposed to exceed the best-match similarity to the stored prototypes of any other task.

The stored and the test prototypes are embedded coherently. Fig. 5 shows, for each dataset, the stored training prototypes $\mathbf{p}_c$ and the test prototypes $\mathbf{q}_c$ of all classes and tasks embedded in a single two-dimensional t-SNE map; hollow markers are the training prototypes and filled markers are the test prototypes, color encodes the task (with shades distinguishing the classes within a task), and the marker shape distinguishes the tasks. Two features stand out. First, the filled marker of a task consistently falls close to the hollow marker of the same task, so the training and test prototypes of a task are embedded together rather than drifting apart; this is precisely the local structure that the nearest-prototype rule of Eq. (5) exploits. Second, several tasks occupy clearly distinct regions



(a) CIFAR-100 (b) ImageNet-R (c) ImageNet-A

Figure 6: Task-identification confusion matrices after the whole task sequence of each dataset has been trained. Each panel reports the $N \times N$ cosine-similarity matrix $\mathbf{S}$ of Eq. (17) between the stored training class prototypes (rows) and the test class prototypes (columns), aggregated per task by averaging the best class-level match; the diagonal is outlined in red. Panels: (a) CIFAR-100 (10 tasks, 10 classes each), (b) ImageNet-R (10 tasks, 20 classes each), and (c) ImageNet-A (10 tasks, 20 classes each). In all three panels every row attains its maximum on the diagonal, i.e., $S_{k,k} > \max_{k' \neq k} S_{k,k'}$, which is exactly the condition of Eq. (18).

of the map (for example, the well-separated task clusters in the ImageNet-A panel), while others remain interleaved, as expected for a nonlinear embedding that preserves local rather than global geometry. We therefore read this map qualitatively, as evidence that the frozen backbone embeds the stored and the test prototypes coherently, and we quantify the actual separation between tasks in the full feature space next.

The similarity matrix confirms the ordering. Fig. 6 reports the matrix $\mathbf{S}$ of Eq. (17) for each dataset, with the diagonal outlined. Two observations follow. First, in every one of the $3N$ rows the diagonal entry is the strictly largest one, so no test prototype of any task matches the stored prototypes of a foreign task better than those of its own task; the dominance holds uniformly over the whole sequence and over all three datasets. Second, the margin δ of Eq. (18) is clearly positive in every row, and its size tracks the difficulty of the benchmark: it is widest on CIFAR-100 (diagonal 0.95–0.96 against off-diagonal values of at most 0.84), narrower on ImageNet-R (diagonal 0.93–0.95), and narrowest on the more fine-grained ImageNet-A (diagonal 0.56–0.87), where semantically related tasks are legitimately closer to one another. This structure is reproduced on CIFAR-100, ImageNet-R, and ImageNet-A despite their very different domains, granularities, and image statistics. Since $\mathbf{S}$ is built from the same frozen-backbone features that PGTR compares at inference, these results provide direct empirical evidence that the frozen backbone places the stored class prototypes so that they remain the nearest reference for the test samples of their own task, which is what lets the logit masking of Eq. (6) be applied to the right class subset and keeps the classifier from competing against classes of other tasks.

After the full sequence has been trained on CIFAR-100, ImageNet-R, and ImageNet-A, PGTR selects the correct task for 100% of the test samples of every task. Task identification thus contributes no error of its own to the final prediction, and the only confusions that can remain are confusions among the classes *within* the routed task, which is consistent with Tab. 5: adding PGTR produces the largest Class-IL improvement.

D Extended Related Work

Rehearsal-Free Class-Incremental Learning. Class-incremental learning (CIL) requires a model to sequentially acquire new classes from a non-stationary data stream while retaining the ability to recognize all previously seen ones, and its central difficulty is the stability–plasticity dilemma: updates that favor new classes tend to overwrite knowledge of old ones. Because historical data is frequently unavailable due to privacy or storage constraints, rehearsal-free CIL has become the predominant practical setting. The advent of large-scale pre-trained models (PTMs) has shifted the field from training from scratch to parameter-efficient fine-tuning (PEFT) of a frozen backbone, where only a small set of injected parameters is adapted per task (He et al. (2025); Wu et al. (2025)). This paradigm inherits the strong generalization of the pre-trained representation and reduces overfitting under limited per-task data, but it also sharpens the efficiency requirement, because the injected pa-

rameters must not grow without bound as tasks accumulate. RE-LoRA targets exactly this regime: it adapts a frozen ViT through a single low-rank adapter whose size is fixed for the entire task sequence.

Prompt-Based Methods. A large family of rehearsal-free CIL methods adapts a frozen ViT through learnable prompt tokens. L2P (Wang et al. (2022b)) maintains a prompt pool and selects instance-specific prompts by key matching; DualPrompt (Wang et al. (2022a)) separates general, task-invariant prompts from expert, task-specific ones; CODA-Prompt (Smith et al. (2023)) replaces discrete key-based selection with an end-to-end attention mechanism over prompt components; and HiDe-Prompt (Wang et al. (2023)) decomposes the CIL objective into within-task prediction, task-identity inference, and task-adaptive prediction to exploit pre-trained knowledge more fully. These approaches do not modify the backbone and are largely rehearsal-free, but they share two structural limitations. First, the prompt pool expands with the number of tasks, which both increases storage and makes the selection of a task-relevant prompt increasingly ambiguous as the sequence lengthens (Lai et al.). Second, inference depends on that selection, so errors in prompt retrieval directly translate into misclassification and add inference-time overhead (Wu et al. (2025)). RE-LoRA avoids an expanding pool altogether by maintaining a single constant-size adapter, and it decouples task identification from feature adaptation through prototypes drawn from the strictly frozen backbone.

Low-Rank Adaptation for Continual Learning. LoRA (Hu et al. (2022)) parameterizes a weight update as the product of two low-rank matrices, attaining strong adaptation capacity with few trainable parameters, and has become a popular substrate for CIL. Existing LoRA-based methods broadly fall into two families. The first preserves historical knowledge by allocating task-specific low-rank components. SD-LoRA (Wu et al. (2025)) separates the learning of the magnitude and direction of LoRA components across tasks; CL-LoRA (He et al. (2025)) couples a task-shared adapter with task-specific adapters and orthogonal block-wise weights; InfLoRA (Liang & Li (2024)) pre-designs a dimensionality-reduction subspace that suppresses interference with historical gradient directions, but relies on stored exemplars to construct it; and BiLoRA (Zhu et al. (2025)) instead expands the parameter space with fixed orthogonal bases so that task subspaces are almost orthogonal with high probability, avoiding explicit interference-elimination optimization. While effective at preserving old knowledge, these methods retain a component for every encountered task or require auxiliary memory, so their parameter or storage footprint grows with the sequence and long-term scalability is limited. The second family keeps a compact shared low-rank update and mitigates interference through regularization or gradient constraints. EWC-LoRA (Zheng et al. (2026)) regularizes a single shared LoRA using a full-dimensional Fisher information matrix, and MoCL (Lai et al.) employs curvature-aware metabolic gating to attenuate updates that conflict with historical knowledge. Such methods hold the parameter count constant, but they introduce additional computation and memory for the importance statistics and, as historical constraints accumulate, progressively restrict the model's capacity to adapt to new tasks. RE-LoRA also operates under a constant budget, but rather than penalizing updates it prevents overwriting structurally through additive residual composition, and it consolidates knowledge by recompression instead of accumulating an ever-growing set of constraints, so no auxiliary regularization term is required.

Model Merging and Low-Rank Consolidation. Model merging integrates knowledge from trained models into a single set of weights without joint retraining, and it has recently been connected to continual learning. LoRA-P&M (Qiu et al. (2025)) forms a convex combination of the previous and the newly trained task model, scaling the task vector by a closed-form coefficient derived from the Hessian, and SD-LoRA (Wu et al. (2025)) can likewise be viewed as a merging scheme that incrementally composes decoupled magnitude and direction components. These methods move toward a unified model, but the merged parameters are obtained either by scalar interpolation or by accumulating additional low-rank components, and no step enforces a fixed rank budget. Historical Knowledge Consolidation in RE-LoRA differs in both respects: it merges the frozen historical adapter and the residual update into a single dense matrix and re-compresses the result to the prescribed rank via truncated SVD, which by the Eckart–Young theorem (Eckart & Young (1936)) is the optimal low-rank approximation of the combined adaptation. This operation reparameterizes one evolving adapter rather than interpolating weights or accumulating components, so the parameter footprints remain constant while the discarded spectral energy is provably minimal.