TIP-LoRA: Two-Sided Interference-Aware Projection for Exemplar-Free Class-Incremental Learning

Anonymous authors
Paper under double-blind review

Abstract

Continual learning (CL) requires a model to acquire new knowledge without forgetting old knowledge, yet low-rank adaptation (LoRA) applied to pre-trained models remains vulnerable to interference from sequential updates. Existing LoRA-based CL methods constrain an update from the input side; with a finite-rank subspace, however, residual output changes can remain, and their interference also depends on the loss response. In this paper, we propose Two-Sided Interference-Aware Projection LoRA (TIP-LoRA), a two-branch framework that controls this interference through paired input- and output-side projections. A local Taylor analysis shows that first-order interference is jointly determined by the layer input and the backpropagated loss response. This analysis suggests controlling both the input- and output-side directions of a new update. Since old-task signals are unavailable, the stability branch estimates these directions from cumulative task updates, and restricts new updates to their two-sided orthogonal complement. In parallel, the plasticity branch learns within the leading singular subspaces of the frozen pre-trained weights to retain transferable structure for new knowledge. We further provide a local bound on the old-task loss change under the projected update. Across seven settings on five benchmarks, TIP-LoRA achieves the highest final accuracy in all seven settings and competitive average incremental accuracy. Controlled diagnostics further show that two-sided projection removes residual interference left by input-only control and that cumulative updates retain the paired directions used by the stability branch.

1 Introduction

Intelligent systems deployed in changing real-world environments, from autonomous agents and robots to medical decision-support systems, must acquire new knowledge while retaining what they have already learned (Parisi et al., 2019; Lesort et al., 2020; Wu et al., 2024). Continual learning (CL) addresses this need by learning from sequential tasks while retaining earlier capabilities (Kirkpatrick et al., 2017; Li & Hoiem, 2016). Within CL, class-incremental learning (CIL) is particularly challenging because new classes arrive successively, while the model must recognize all observed classes without task identity at test time (Wang et al., 2022b;a; Smith et al., 2023). This shared classifier must accommodate an expanding set of classes, so updates for the current task can interfere with knowledge learned from earlier tasks. Replaying old examples reduces this interference (Rebuffi et al., 2017), but memory and privacy constraints may prevent their storage. When replay is unavailable, exemplar-free CIL must learn from current-task data alone, so each update must limit interference while remaining flexible enough to learn new classes.

With the rise of large-scale pre-trained models, recent exemplar-free CIL methods increasingly address this balance by freezing the transferable backbone and adapting it with only a small set of trainable parameters. These parameter-efficient fine-tuning (PEFT) techniques include prompt pools, lightweight adapters, and low-rank adaptation (LoRA) modules (Wang et al., 2022b; Smith et al., 2023; Tan et al., 2024; Hu et al., 2022). Recent LoRA-based CIL methods construct the adaptation subspace from layer-input features to reduce interference with earlier tasks achieving remarkable CL performances. (Liang & Li, 2024; Liu & Chang, 2025; Li et al., 2026; He et al., 2026).

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.

These LoRA-based methods reduce interference by limiting the action of an update on selected old-task input feature directions and thereby reducing the corresponding layer-output changes (Saha et al., 2021; Liang & Li, 2024; Li et al., 2026). However, the protected feature subspace is estimated at finite rank, so some old-task directions can remain outside the projection. Updates along those uncovered directions can still change the layer output, and related null-space work shows that the resulting residual can propagate through subsequent nonlinear layers (Kang et al., 2025). Even after input-side control, however, the effect of a remaining output change still depends on how the downstream loss responds to its direction.

To formalize this remaining interference, we examine how a layer-weight update changes an old-task loss. A first-order Taylor expansion shows that the leading loss change depends jointly on two signals: the layer input determines how the update changes the layer output, whereas the backpropagated loss response determines how that output change affects the loss. Input-side projection controls only the layer input. TIP-LoRA therefore also projects the stability update on the output side. After the two projections, its first-order effect depends only on the remaining components of the layer input and loss response. The resulting stability branch is illustrated in Figure 1(a).

Constructing these two projectors from old-task features and loss responses would require unavailable old data. TIP-LoRA instead uses the optimization history already stored in the model. For a linear map, each weight gradient is an outer product of a loss response and a layer input, so response–feature pairs shared during training can leave paired left and right structure in the task updates. TIP-LoRA accumulates these updates and extracts their leading singular directions as the stability reference; the next stability update is restricted to the corresponding left and right orthogonal complements. Since this constraint excludes directions summarized by earlier task updates, the plasticity branch uses directions already encoded by the frozen pre-trained weights to exploit transferable structure from pre-training. The two branches therefore use distinct data-free references for retention and current-task learning. Across five benchmarks, TIP-LoRA achieves competitive performance. We further use old-signal controls to examine the two-sided interference mechanism, historical-update diagnostics to evaluate the stability reference, and branch ablations to examine the complete design.

Our main contributions are:

- We show that the first-order change in an old-task loss depends jointly on the layer input and the backpropagated loss response, motivating two-sided projection.
- We propose TIP-LoRA, which uses cumulative task updates to construct a two-sided stability constraint and pre-trained-weight singular subspaces to support plasticity, without storing old-task examples.
- Across seven settings on five benchmarks, TIP-LoRA achieves the competitive performance in all settings; controlled studies further verify the two branches and output-side projection.

2 Related Work

Continual learning. Continual learning seeks to acquire new knowledge without forgetting earlier tasks. Representative approaches preserve past knowledge through regularization, replay, or parameter isolation. Elastic weight consolidation (EWC) and synaptic intelligence regularize important parameters (Kirkpatrick et al., 2017; Zenke et al., 2017). Learning without Forgetting distills the previous model on new-task data (Li & Hoiem, 2016), while iCaRL retains exemplars (Rebuffi et al., 2017) and PackNet allocates parameters through pruning (Mallya & Lazebnik, 2018).

Continual learning with pre-trained models. PEFT builds on pre-trained representations while limiting the parameters learned for each task. L2P and DualPrompt learn prompt pools, and CODA-Prompt composes prompts through attention (Wang et al., 2022b;a; Smith et al., 2023). HiDe-Prompt decomposes prompt-based learning, EvoPrompt evolves a parameterized prompt memory, and Learning-Accumulation-Ensemble (LAE) provides a unified framework for parameter-efficient tuning (Wang et al., 2023; Kurniawan et al., 2024; Gao et al., 2023). LoRA instead learns low-rank weight updates (Hu et al., 2022). Its extensions to CL include SD-LoRA, which separates direction and magnitude learning; PLAN, which proactively allocates low-rank components (Wu et al., 2025; Liu & Chang, 2025; Wang et al., 2025). LoRA-P&M combines perturbed training with model merging (Qiu et al., 2025). More recently, SplitLoRA selects minor gradient subspaces to balance stability

and plasticity (Qiu et al., 2026), while EWC-LoRA regularizes the composite low-rank update using accumulated Fisher information (Zheng et al., 2026). BiLoRA uses a bilinear construction with orthogonal bases to organize task-specific parameter spaces (Zhu et al., 2025). These methods highlight that reducing trainable parameters alone does not prevent forgetting; controlling the learned updates is equally important.

Projection and subspace-based adaptation. Gradient Projection Memory (GPM) uses activation representations to construct compact layer-wise gradient constraints (Saha et al., 2021). InfLoRA connects low-rank adaptation to constrained weight optimization and constructs its subspace from layer-input features (Liang & Li, 2024). LoRA-DRS constructs a drift-resistant subspace from current-task input features obtained after subtracting earlier LoRA updates (Liu & Chang, 2025). DGS constructs a subspace from layer-input features of the frozen pre-trained model and combines the original gradients with their projections onto this subspace (Li et al., 2026). These methods efficiently capture input-side structure. DMNSP uses a common subspace across layer-wise approximate null spaces to reduce the propagation of nonzero feature residuals (Kang et al., 2025). With a finite subspace, the remaining output change can still affect an old-task loss through downstream response.

3 PRELIMINARIES

Problem setting. We consider exemplar-free CIL (Liang & Li, 2024) with a sequence of T tasks. Task t provides $\mathcal{D}_t = \{(\mathbf{x}_{i,t}, y_{i,t})\}_{i=1}^{n_t}$, whose labels belong to a class set $\mathcal{C}_t$ disjoint from those of the other tasks. During task t, only $\mathcal{D}_t$ is available; after training, the model predicts over $\bigcup_{j=1}^{t} \mathcal{C}_j$ without access to the task identity. We adapt a pre-trained Vision Transformer (ViT) with LoRA modules; the resulting feature extractor and classifier are denoted by f_t and c_t, respectively.

Low-rank adaptation. For an adapted linear map in the backbone, let $\mathbf{W}_{\mathrm{pt}} \in \mathbb{R}^{d_{\mathrm{out}} \times d_{\mathrm{in}}}$ be its frozen pre-trained weight (PTW), with input and output dimensions d_{in} and d_{out}. LoRA (Hu et al., 2022) freezes $\mathbf{W}_{\mathrm{pt}}$ and parameterizes its update as

$$\mathbf{z} = \mathbf{W}_{\mathrm{pt}}\mathbf{h} + \Delta\mathbf{W}\mathbf{h}, \qquad \Delta\mathbf{W} = \frac{\alpha}{r}\mathbf{B}\mathbf{A}, \tag{1}$$

where $\mathbf{h} \in \mathbb{R}^{d_{\mathrm{in}}}$ and $\mathbf{z} \in \mathbb{R}^{d_{\mathrm{out}}}$ are the layer input and output. The trainable factors are $\mathbf{A} \in \mathbb{R}^{r \times d_{\mathrm{in}}}$ and $\mathbf{B} \in \mathbb{R}^{d_{\mathrm{out}} \times r}$; $r > 0$ is a small integer rank and α is a scaling coefficient. Our analysis applies to any adapted linear map, so we omit layer indices. We use $\|\cdot\|_2$ for vector Euclidean and matrix spectral norms, $\|\cdot\|_F$ for the Frobenius norm, and $\top$ for transpose.

Feature-based projection. GPM and InfLoRA summarize observed layer inputs in feature subspaces (Saha et al., 2021; Liang & Li, 2024). Let $\mathbf{P}_h \in \mathbb{R}^{d_{\mathrm{in}} \times d_{\mathrm{in}}}$ be the orthogonal projector onto the k_h leading left-singular directions of stacked input columns. Updates satisfying $\Delta\mathbf{W}\mathbf{P}_h = 0$ preserve the selected span and give

$$\Delta\mathbf{W}\mathbf{h}_s = \Delta\mathbf{W}(\mathbf{I} - \mathbf{P}_h)\mathbf{h}_s \tag{2}$$

Here $\mathbf{h}_s$ is an old-task input and $\mathbf{I} \in \mathbb{R}^{d_{\mathrm{in}} \times d_{\mathrm{in}}}$ is the identity matrix. For inputs in the selected span, the layer output and its downstream response remain unchanged with the other parameters fixed.

4 METHOD

TIP-LoRA begins with a simple question: how can a new weight update change an old-task loss? We show that the answer involves two signals at the adapted layer: its input and the loss response at its output. This two-signal view identifies which directions can produce first-order interference. TIP-LoRA therefore projects the stability update orthogonally to these directions on both sides of the weight matrix. Because old samples are unavailable, we estimate the directions from the weight updates left by earlier tasks. We then use a fixed PTW basis as a separate route for learning the current task. Figure 1(a) presents the complete architecture. We first establish the two-sided interference principle illustrated in panel (c), then construct the historical projectors summarized in panel (b), and finally return to panel (a) to assemble the two-branch update.

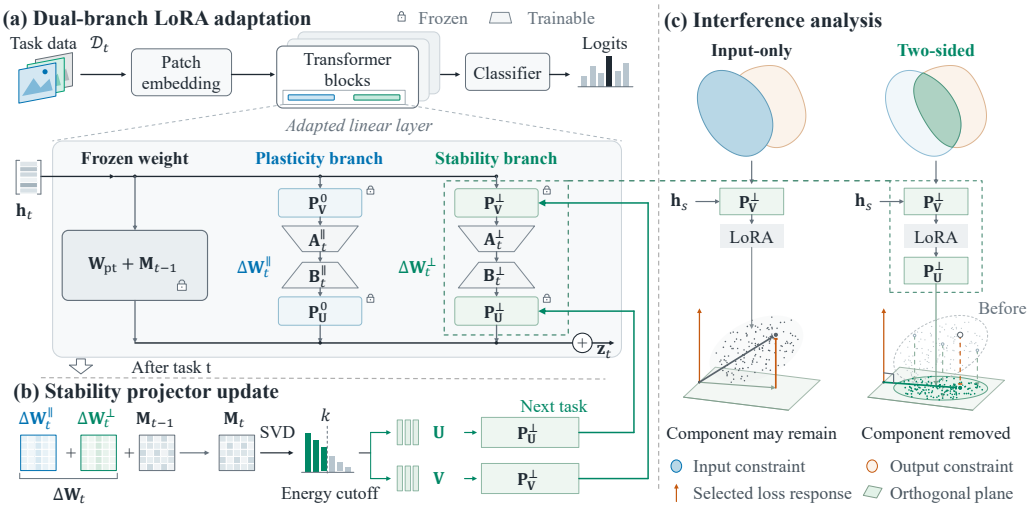


Figure 1: Overview of TIP-LoRA. (a) A frozen backbone is adapted by two projected low-rank branches. The plasticity branch uses fixed PTW singular subspaces, while the stability branch uses the complements of bases obtained from historical updates. (b) After each task, the two realized branch updates are merged into the historical update matrix; its truncated singular value decomposition (SVD) refreshes the stability projectors for the next task. (c) Input-only projection can leave component to a selected loss-response direction, whereas two-sided projection removes that selected component.

4.1 Two-Sided Interference

Consider an old-task sample s and a linear map $\mathbf{z}_s = \mathbf{W}\mathbf{h}_s$ inside the network.[1] Here $\mathbf{h}_s \in \mathbb{R}^{d_{\text{in}}}$ and $\mathbf{z}_s \in \mathbb{R}^{d_{\text{out}}}$ are the layer input and output, and $\mathbf{W} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ is the reference weight. We perturb this weight by $\Delta\mathbf{W}$ while holding the input and the other parameters fixed. Let ℓ_s be the loss of the network on sample s, and define

$$\mathbf{e}_s = \nabla_{\mathbf{z}_s} \ell_s \in \mathbb{R}^{d_{\text{out}}}. \tag{3}$$

Thus $\mathbf{e}_s$ is the loss gradient backpropagated to this layer output. A weight update changes the output by $\Delta\mathbf{W}\mathbf{h}_s$, and we use $\mathbf{e}_s$ to quantify the resulting local loss change. Let $\delta\ell_s(\Delta\mathbf{W})$ denote the change in the old-task loss after applying $\Delta\mathbf{W}$. The following local condition allows us to separate its leading change from the higher-order remainder.

Assumption 1 (Local smoothness)**.** *With $\mathbf{h}_s$ and all other parameters fixed, the old-task loss has an L_s-Lipschitz gradient with respect to the layer output on a convex neighborhood of $\mathbf{z}_s$ that contains $\mathbf{z}_s + \Delta\mathbf{W}\mathbf{h}_s$. The finite constant $L_s \geq 0$ is fixed on this neighborhood.*

Under Assumption 1, a first-order Taylor expansion gives

$$\delta\ell_s(\Delta\mathbf{W}) = \mathbf{e}_s^\top \Delta\mathbf{W}\mathbf{h}_s + R_s, \qquad |R_s| \leq \frac{L_s}{2}\|\Delta\mathbf{W}\mathbf{h}_s\|_2^2. \tag{4}$$

Appendix A.1 gives the complete derivation of Eq. (4). Its leading term makes the two roles explicit. The input $\mathbf{h}_s$ determines the output change produced by $\Delta\mathbf{W}$, and $\mathbf{e}_s$ determines how the loss responds to that change. This form also shows that controlling only one of the two signals leaves the other one free.

The input-side construction in Eq. (2) constrains the update on the right. Writing $\mathbf{P}_V^{\perp} = \mathbf{I} - \mathbf{P}_V$ and $\bar{\mathbf{h}}_s = \mathbf{P}_V^{\perp}\mathbf{h}_s$, applying that construction replaces a candidate update by $\Delta\mathbf{W}\mathbf{P}_V^{\perp}$ and changes the leading term to $\mathbf{e}_s^\top \Delta\mathbf{W}\bar{\mathbf{h}}_s$. The selected feature component is removed, but the remaining feature still interacts with the full loss response. Existing multi-layer null-space work identifies the same limitation: a finite-rank input projector can leave a nonzero feature residual that propagates through subsequent layers (Kang et al., 2025).

[1]For a weight shared across tokens, the weight gradient sums the token contributions. Appendix A.5 gives the corresponding matrix form.

The remaining interaction points to the missing constraint: the loss response must also be controlled. TIP-LoRA therefore selects an orthogonal output-side projector $\mathbf{P}_U$. With $\mathbf{P}_U^{\perp} = \mathbf{I} - \mathbf{P}_U$, it projects the candidate update on both sides,

$$\Delta\mathbf{W}_{\perp} = \mathbf{P}_U^{\perp}\Delta\mathbf{W}\mathbf{P}_V^{\perp}. \tag{5}$$

Here $\Delta\mathbf{W}$ is the candidate update before projection. Because the projectors are orthogonal, the resulting update has no component in the selected output or input subspace. Defining $\bar{\mathbf{e}}_s = \mathbf{P}_U^{\perp}\mathbf{e}_s$, its first-order effect is

$$\mathbf{e}_s^{\top}\Delta\mathbf{W}_{\perp}\mathbf{h}_s = \bar{\mathbf{e}}_s^{\top}\Delta\mathbf{W}\bar{\mathbf{h}}_s. \tag{6}$$

Thus the right constraint removes the selected input component, while the left constraint removes the selected loss-response component. Only their residual interaction remains. Figure 1(c) contrasts the input-only and two-sided constructions.

Proposition 1 (Two-sided residual interference)**.** *Let $\mathbf{P}_U$ and $\mathbf{P}_V$ be orthogonal projectors. The stability update in Eq.* (5) *obeys Eq.* (6) *and*

$$\left|\mathbf{e}_s^{\top}\Delta\mathbf{W}_{\perp}\mathbf{h}_s\right| \le \|\Delta\mathbf{W}\|_2\|\bar{\mathbf{e}}_s\|_2\|\bar{\mathbf{h}}_s\|_2. \tag{7}$$

Equation (7) is the main design result: after both projections, first-order interference depends only on the two residuals. The projectors are therefore chosen with respect to the two signals that determine interference; we refer to this construction as two-sided interference-aware projection. Appendix A.4 extends the result to a finite update by including the quadratic Taylor remainder. Having established what the two projectors should protect, we next explain how to obtain them without old data.

4.2 Constructing Stability Projectors from Historical Updates

Gradient structure of historical updates. We now turn to the two orthogonal bases required above. Returning to the same linear map, the chain rule gives the exact weight-gradient factorization

$$\nabla_{\mathbf{W}}\ell_s(\mathbf{W}\mathbf{h}_s) = \mathbf{e}_s\mathbf{h}_s^{\top}. \tag{8}$$

This equation supplies the missing connection: every sample gradient pairs one loss-response direction on the left with one input-feature direction on the right. If old data were available, these direction pairs could be estimated by aggregating their outer products. Exemplar-free learning removes that option, but the updates produced while learning the old tasks remain available.

Let $\Delta\mathbf{W}_\tau$ denote the LoRA update merged after task τ. We summarize the realized updates from all earlier tasks as

$$\mathbf{M}_{t-1} = \sum_{\tau=1}^{t-1}\Delta\mathbf{W}_\tau = -\sum_{q\in\mathcal{Q}_{<t}}\eta_q\mathbf{e}_q\mathbf{h}_q^{\top} + \mathbf{R}_{t-1}^{\mathrm{opt}}. \tag{9}$$

Figure 1(b) summarizes this merge-and-refresh cycle: the two branch updates are merged with the preceding history, and the SVD of the resulting matrix supplies the next pair of stability projectors. Here $\mathcal{Q}_{<t}$ indexes historical gradient contributions, η_q is the corresponding step weight, and $\mathbf{R}_{t-1}^{\mathrm{opt}}$ is the exact difference between their ideal first-order sum and the realized LoRA updates. Equation (8) explains the second form. Existing work on gradient coherence shows that gradient components shared by many examples become stronger when gradients are summed (Chatterjee, 2020; Zielinski et al., 2020). Accordingly, response–feature pairs that recur with a similar orientation accumulate in $\mathbf{M}_{t-1}$, whereas inconsistent pairs tend to cancel. The leading structure of $\mathbf{M}_{t-1}$ therefore emphasizes pairs reinforced during earlier training, while $\mathbf{R}_{t-1}^{\mathrm{opt}}$ captures the low-rank parameterization and optimizer.

Retained historical structure. We now separate the consistently reinforced directions from the remaining variation. Write each loss response and feature as

$$\mathbf{e}_s = \mathbf{U}_\star\mathbf{a}_s + \mathbf{r}_s^e, \qquad \mathbf{h}_s = \mathbf{V}_\star\mathbf{b}_s + \mathbf{r}_s^h, \tag{10}$$

where $\mathbf{U}_\star$ and $\mathbf{V}_\star$ span the dominant historical response and feature directions, $\mathbf{a}_s$ and $\mathbf{b}_s$ are their coordinates, and the two residuals contain the remaining variation. The residuals are orthogonal to their respective subspaces.

Substituting these decompositions into Eq. (9) and summing separates the component supported jointly on both dominant subspaces, including any such component of $\mathbf{R}_{t-1}^{\mathrm{opt}}$, from all terms containing at least one residual. Hence,

$$\mathbf{M}_{t-1} = \mathbf{U}_\star \mathbf{C}_{t-1} \mathbf{V}_\star^\top + \mathbf{E}_{t-1}. \tag{11}$$

Here $\mathbf{C}_{t-1}$ records the accumulated coupling between the two dominant historical subspaces, while $\mathbf{E}_{t-1}$ contains the remaining variation. Appendix A.2 gives the full four-term decomposition.

The SVD can identify the paired component only if it is separated from this residual. We state that requirement explicitly.

Assumption 2 (Retained historical component)**.** *In Eq.* (11)*,* $\sigma_k(\mathbf{C}_{t-1}) = \gamma_{t-1} > 0$ *and* $\|\mathbf{E}_{t-1}\|_2 \leq \varepsilon_{t-1} < \gamma_{t-1}/2$.

Here γ_{t-1} is the weakest retained paired mode and ε_{t-1} bounds the remaining variation. Their separation gives the rank-k spectral gap used below. Appendix A.2 gives a sufficient bound for ε_{t-1}, and Appendix A.3 proves the resulting recovery guarantee.

Recovering the stability projectors. At task t, we obtain both bases and their projectors from the rank-k truncated SVD of the historical update matrix:

$$\mathrm{SVD}_k(\mathbf{M}_{t-1}) = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \qquad \mathbf{P}_U = \mathbf{U}\mathbf{U}^\top, \quad \mathbf{P}_V = \mathbf{V}\mathbf{V}^\top. \tag{12}$$

Under Assumption 2, singular-subspace perturbation gives

$$\max\{\|\mathbf{P}_U^\perp \mathbf{U}_\star\|_2, \|\mathbf{P}_V^\perp \mathbf{V}_\star\|_2\} \leq \frac{\varepsilon_{t-1}}{\gamma_{t-1} - \varepsilon_{t-1}}. \tag{13}$$

Appendix A.3 proves the bound, and a cumulative-energy threshold selects the rank in practice. The bound gives the desired bridge. When the retained component has the stated margin, the leading left and right singular vectors of $\mathbf{M}_{t-1}$ stay close to the dominant historical response and feature subspaces. We can therefore use them as the two projectors in Proposition 1.

Initialization and stability update. The construction above applies once a historical update exists. At the first task, however, $\mathbf{M}_0 = \mathbf{0}$. We therefore initialize the two bases from the PTW SVD,

$$\mathbf{W}_{\mathrm{pt}} = \mathbf{U}^0 \mathbf{\Sigma}^0 (\mathbf{V}^0)^\top, \qquad \mathbf{P}_U^0 = \mathbf{U}^0 (\mathbf{U}^0)^\top, \qquad \mathbf{P}_V^0 = \mathbf{V}^0 (\mathbf{V}^0)^\top, \tag{14}$$

where $\mathbf{U}^0$ and $\mathbf{V}^0$ contain the leading PTW vectors selected by the same energy rule. For task one, the stability branch uses the complements of these projectors. Once task one is merged, Eq. (12) replaces the initialization with bases estimated from historical updates. For task t, the stability branch is

$$\Delta\mathbf{W}_t^\perp = \frac{\alpha_\perp}{r_\perp} \mathbf{P}_U^\perp \mathbf{B}_t^\perp \mathbf{A}_t^\perp \mathbf{P}_V^\perp. \tag{15}$$

This formula instantiates Eq. (5) with $\Delta\mathbf{W} = (\alpha_\perp / r_\perp)\mathbf{B}_t^\perp \mathbf{A}_t^\perp$. Consequently, the branch has zero Frobenius inner product with the retained rank-k reconstruction of $\mathbf{M}_{t-1}$. More importantly, Proposition 1 controls its local coupling to the retained response and feature directions.

4.3 Plasticity and Continual Update

With the stability branch now specified, we return to the complete architecture in Figure 1(a). Removing selected directions protects earlier tasks, but it can also reduce the capacity available to the current task. TIP-LoRA therefore adds a plasticity branch defined by the retained PTW directions. Because the PTW projectors in Eq. (14) remain fixed, they provide the same exploration coordinates throughout learning:

$$\Delta\mathbf{W}_t^\parallel = \frac{\alpha_\parallel}{r_\parallel} \mathbf{P}_U^0 \mathbf{B}_t^\parallel \mathbf{A}_t^\parallel \mathbf{P}_V^0. \tag{16}$$

The plasticity branch therefore searches within high-energy pre-trained directions, while the stability branch follows the complementary constraint estimated from historical updates. Their roles are distinct, but their updates are merged into the same adapted weight.

For branch $j \in \{\parallel, \perp\}$, $\mathbf{A}_t^j \in \mathbb{R}^{r_j \times d_{\mathrm{in}}}$ and $\mathbf{B}_t^j \in \mathbb{R}^{d_{\mathrm{out}} \times r_j}$ are trainable LoRA factors, r_j is the branch rank, and α_j is its scale. The realized task update and its merge are

$$\Delta\mathbf{W}_t = \Delta\mathbf{W}_t^\parallel + \Delta\mathbf{W}_t^\perp, \qquad \mathbf{M}_t = \mathbf{M}_{t-1} + \Delta\mathbf{W}_t, \qquad \mathbf{W}_t = \mathbf{W}_{\mathrm{pt}} + \mathbf{M}_t. \tag{17}$$

This merge also updates the historical matrix in Eq. (9). Its SVD then supplies the two stability bases for task $t + 1$, which closes the update cycle.

No additional loss is required to enforce this construction because the constraints are built into the two parameterizations. The projected branches are optimized jointly using standard cross-entropy.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Datasets and protocol. We evaluate on five standard CIL benchmarks: ImageNet-R (Hendrycks et al., 2021a), CIFAR-100 (Krizhevsky et al., 2009), CUB-200 (Wah et al.), ImageNet-A (Hendrycks et al., 2021b), and DomainNet (Peng et al., 2019). Following common CIL protocols (Gao et al., 2023; Liang & Li, 2024; Wu et al., 2025), ImageNet-R is divided into 5, 10, or 20 tasks with 40, 20, or 10 classes per task. CIFAR-100, CUB-200, and ImageNet-A use 10 tasks with 10, 20, and 20 classes per task, respectively; DomainNet uses 5 tasks with 69 classes per task. Training uses only current-task images. At test time, predictions cover all seen classes without task identities.

Implementation. We use ViT-B/16 pre-trained on ImageNet-21K and subsequently fine-tuned on ImageNet-1K. Query (Q) and value (V) weights are adapted in all 12 blocks. Training uses SGD, batch size 48, and cosine decay. We train for 20 epochs per task on all benchmarks except DomainNet, where we use 5 epochs. Learning rates and branch ranks are listed in Appendix B.1. We train the normalized cosine classifier using standard current-task cross-entropy. Appendix B.1 provides the complete objective, benchmark configurations, and source conventions.

Metrics. We use two widely adopted CIL metrics (Gao et al., 2023; Liang & Li, 2024; Wu et al., 2025). Let A_t denote accuracy over all seen classes after task t, and let T be the number of tasks. Final accuracy is A_T, and average incremental accuracy is $\bar{A} = T^{-1} \sum_{t=1}^{T} A_t$; higher is better for both. For the architecture ablation, we additionally report final forgetting (Chaudhry et al., 2018). If $a_{t,i}$ is the accuracy on task i after learning task t, it is $F_T = (T-1)^{-1} \sum_{i=1}^{T-1} (\max_{i \le t < T} a_{t,i} - a_{T,i})$; lower is better. Accuracies are percentages. Each benchmark entry retains the uncertainty convention reported by its source.

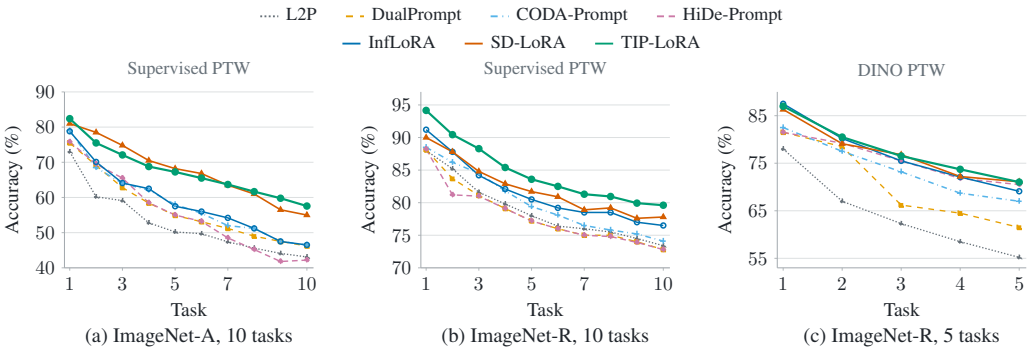
Comparison methods. Following prior ViT-based CIL studies (Liang & Li, 2024; Wu et al., 2025), we compare with representative prompt-based, feature-projection, and LoRA methods. Tables 1 and 2 include results reported under the corresponding dataset, task split, and backbone protocol.

Table 1: ImageNet-R at three task lengths. Accuracy (%); TIP-LoRA reports the mean and standard deviation over three runs, while baseline uncertainties follow their sources. Best results are bold and second-best results are underlined. FT denotes full fine-tuning;

Method	$T = 5$		$T = 10$		$T = 20$	
	$A_T \uparrow$	$\bar{A} \uparrow$	$A_T \uparrow$	$\bar{A} \uparrow$	$A_T \uparrow$	$\bar{A} \uparrow$
FT	$18.74_{\pm 0.44}$	$48.39_{\pm 0.58}$	$10.12_{\pm 0.51}$	$35.23_{\pm 0.92}$	$4.75_{\pm 0.40}$	$22.80_{\pm 0.37}$
L2P (Wang et al., 2022b)	$73.04_{\pm 0.71}$	$76.94_{\pm 0.41}$	$71.26_{\pm 0.44}$	$76.13_{\pm 0.46}$	$68.97_{\pm 0.51}$	$74.16_{\pm 0.32}$
L2P++ (Wang et al., 2022b)	$70.83_{\pm 0.58}$	$78.34_{\pm 0.47}$	$69.29_{\pm 0.73}$	$78.30_{\pm 0.69}$	$65.89_{\pm 1.30}$	$77.15_{\pm 0.65}$
Deep L2P++ (Wang et al., 2022b)	$73.93_{\pm 0.37}$	$80.14_{\pm 0.54}$	$71.66_{\pm 0.64}$	$79.63_{\pm 0.90}$	$68.42_{\pm 1.20}$	$78.68_{\pm 1.03}$
DualPrompt (Wang et al., 2022a)	$69.99_{\pm 0.57}$	$72.24_{\pm 0.41}$	$68.22_{\pm 0.20}$	$73.81_{\pm 0.39}$	$65.23_{\pm 0.45}$	$71.30_{\pm 0.16}$
CODA-Prompt (Smith et al., 2023)	$76.63_{\pm 0.27}$	$80.30_{\pm 0.28}$	$74.05_{\pm 0.41}$	$78.14_{\pm 0.39}$	$69.38_{\pm 0.33}$	$73.95_{\pm 0.63}$
HiDe-Prompt (Wang et al., 2023)	$74.77_{\pm 0.25}$	$78.15_{\pm 0.24}$	$74.65_{\pm 0.14}$	$78.46_{\pm 0.18}$	$73.59_{\pm 0.19}$	$77.93_{\pm 0.19}$
EvoPrompt (Kurniawan et al., 2024)	$77.16_{\pm 0.18}$	$82.22_{\pm 0.54}$	$76.83_{\pm 0.08}$	$\underline{82.09_{\pm 0.68}}$	$74.41_{\pm 0.23}$	$\underline{80.96_{\pm 1.42}}$
InfLoRA (Liang & Li, 2024)	$76.95_{\pm 0.23}$	$81.81_{\pm 0.14}$	$74.75_{\pm 0.64}$	$80.67_{\pm 0.55}$	$69.89_{\pm 0.56}$	$76.68_{\pm 0.57}$
BiLoRA (Zhu et al., 2025)	$78.32_{\pm 0.37}$	$\underline{83.31_{\pm 0.52}}$	$\underline{77.95_{\pm 0.14}}$	$81.52_{\pm 0.26}$	$72.41_{\pm 0.65}$	$79.28_{\pm 0.76}$
PLAN (Wang et al., 2025)	$77.79_{\pm 0.24}$	$81.93_{\pm 0.63}$	$75.25_{\pm 0.42}$	$80.41_{\pm 0.56}$	$71.06_{\pm 0.42}$	$77.93_{\pm 0.56}$
SD-LoRA (Wu et al., 2025)	$\underline{79.15_{\pm 0.20}}$	$83.01_{\pm 0.42}$	$77.34_{\pm 0.35}$	$82.04_{\pm 0.24}$	$\underline{75.26_{\pm 0.37}}$	$80.22_{\pm 0.72}$
SD-LoRA-RR (Wu et al., 2025)	$79.01_{\pm 0.26}$	$82.50_{\pm 0.38}$	$77.18_{\pm 0.39}$	$81.74_{\pm 0.24}$	$74.05_{\pm 0.51}$	$80.65_{\pm 0.35}$
SD-LoRA-KD (Wu et al., 2025)	$78.85_{\pm 0.29}$	$82.47_{\pm 0.58}$	$77.03_{\pm 0.67}$	$81.52_{\pm 0.26}$	$74.12_{\pm 0.66}$	$80.11_{\pm 0.75}$
EWC-LoRA (Zheng et al., 2026)	$76.36_{\pm 0.21}$	$81.43_{\pm 0.13}$	$72.86_{\pm 0.79}$	$78.95_{\pm 0.86}$	$70.18_{\pm 1.06}$	$77.06_{\pm 0.54}$
TIP-LoRA	$\mathbf{80.98}_{\pm 0.33}$	$\mathbf{85.14}_{\pm 0.45}$	$\mathbf{79.61}_{\pm 0.40}$	$\mathbf{84.62}_{\pm 0.67}$	$\mathbf{76.08}_{\pm 0.13}$	$\mathbf{82.21}_{\pm 0.28}$

Table 2: Accuracy across four datasets. Accuracy (%); ten tasks for CIFAR-100, CUB-200 and ImageNet-A, and five for DomainNet. Best results are bold and second-best results are underlined; "–" denotes an unreported result.

Method	CIFAR-100		CUB-200		ImageNet-A		DomainNet	
	$A_T \uparrow$	$\bar{A} \uparrow$	$A_T \uparrow$	$\bar{A} \uparrow$	$A_T \uparrow$	$\bar{A} \uparrow$	$A_T \uparrow$	$\bar{A} \uparrow$
L2P (Wang et al., 2022b)	$83.18_{\pm 1.20}$	$87.69_{\pm 1.05}$	$65.18_{\pm 2.49}$	$76.12_{\pm 1.27}$	$42.94_{\pm 1.27}$	$51.40_{\pm 1.95}$	$70.26_{\pm 0.25}$	$75.83_{\pm 0.98}$
DualPrompt (Wang et al., 2022a)	$81.48_{\pm 0.86}$	$86.41_{\pm 0.66}$	$68.00_{\pm 1.06}$	$79.40_{\pm 0.88}$	$45.49_{\pm 0.96}$	$54.68_{\pm 1.24}$	$68.26_{\pm 0.90}$	$73.84_{\pm 0.45}$
CODA-Prompt (Smith et al., 2023)	$86.31_{\pm 0.12}$	$90.67_{\pm 0.22}$	$71.92_{\pm 0.33}$	$78.76_{\pm 0.65}$	$45.36_{\pm 0.78}$	$57.03_{\pm 0.94}$	$70.58_{\pm 0.53}$	$76.68_{\pm 0.44}$
HiDe-Prompt (Wang et al., 2023)	–	–	–	–	$42.70_{\pm 0.60}$	$56.32_{\pm 0.40}$	$72.20_{\pm 0.08}$	$77.01_{\pm 0.04}$
InfLoRA (Liang & Li, 2024)	$86.75_{\pm 0.35}$	$91.72_{\pm 0.15}$	$70.82_{\pm 0.23}$	$81.39_{\pm 0.14}$	$49.20_{\pm 1.12}$	$60.92_{\pm 0.61}$	$71.59_{\pm 0.23}$	$78.29_{\pm 0.50}$
SD-LoRA (Wu et al., 2025)	$\underline{88.01}_{\pm 0.31}$	$\underline{92.54}_{\pm 0.18}$	$\underline{77.48}_{\pm 0.20}$	$\underline{85.59}_{\pm 0.44}$	$\underline{55.96}_{\pm 0.73}$	$\underline{64.95}_{\pm 1.63}$	$\underline{72.82}_{\pm 0.37}$	$\mathbf{78.89}_{\pm 0.50}$
SD-LoRA-RR (Wu et al., 2025)	$87.26_{\pm 0.22}$	$92.05_{\pm 0.31}$	$76.35_{\pm 0.28}$	$83.89_{\pm 0.35}$	$55.59_{\pm 1.08}$	$64.59_{\pm 1.91}$	$72.58_{\pm 0.40}$	$\underline{78.79}_{\pm 0.78}$
SD-LoRA-KD (Wu et al., 2025)	$87.09_{\pm 0.45}$	$92.01_{\pm 0.33}$	$75.95_{\pm 0.55}$	$83.21_{\pm 0.31}$	$54.24_{\pm 1.12}$	$63.89_{\pm 0.58}$	$72.15_{\pm 0.50}$	$78.44_{\pm 0.66}$
TIP-LoRA	$\mathbf{88.44}_{\pm 0.63}$	$\mathbf{92.82}_{\pm 0.46}$	$\mathbf{78.95}_{\pm 0.23}$	$\mathbf{86.20}_{\pm 1.23}$	$\mathbf{57.56}_{\pm 0.47}$	$\mathbf{67.42}_{\pm 2.25}$	$\mathbf{73.05}_{\pm 0.16}$	$78.81_{\pm 0.34}$

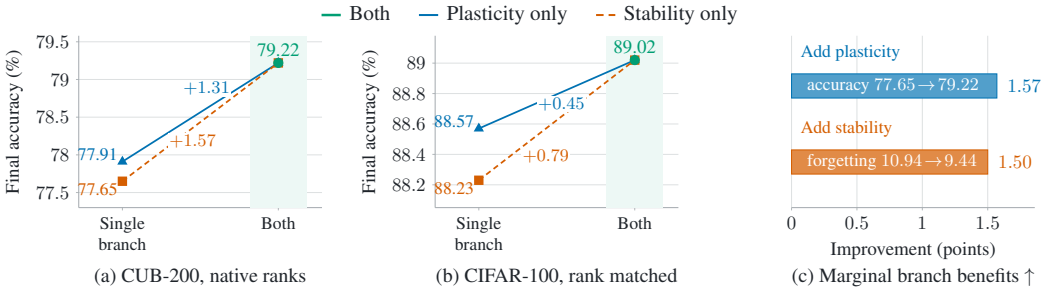


(a) ImageNet-A, 10 tasks (b) ImageNet-R, 10 tasks (c) ImageNet-R, 5 tasks

Figure 2: Task-wise accuracy on ImageNet-A and ImageNet-R under supervised and DINO self-supervised pre-training (Caron et al., 2021).

5.2 Benchmark Performance

Tables 1–2 summarize the main comparison. On ImageNet-R, TIP-LoRA ranks first in both metrics for all three task lengths, with gains of 0.82–2.53 points over the strongest competing entry. It also gives the best final and average accuracy on CIFAR-100 and CUB-200. On ImageNet-A, it improves SD-LoRA by 1.60/2.47 points in final/average accuracy. These results show TIP-LoRA's advantage.



(a) CUB-200, native ranks (b) CIFAR-100, rank matched (c) Marginal branch benefits ↑

Figure 3: Branch cooperation on CUB-200 and CIFAR-100. Panel (b) controls total rank; panel (c) reports the CUB-200 accuracy and forgetting improvements from adding the complementary branch.

Figure 2 shows how these summary scores arise over the task sequence. TIP-LoRA finishes first in both supervised-pretraining panels, with its ImageNet-A advantage emerging mainly in later tasks; under DINO, its final is essentially tied with SD-LoRA's. Appendix B.3 provides the task-wise protocols. To test robustness to the choice of k, we vary the SVD energy threshold: both accuracy measures change by less than 0.4 points, although forgetting varies more (Appendix D.3).

Computation, parameters, and storage. On ten-task ImageNet-R, TIP-LoRA requires 35.12 billion floating-point operations (GFLOPs) per inference, matching InfLoRA and SD-LoRA and half the 70.14–70.61 GFLOPs reported for three prompt-based baselines. It adds 0.442M LoRA parameters per task (versus 0.369M for the two LoRA baselines) and stores no old-task features.

Table 3: Efficiency and accuracy on ten-task ImageNet-R. GFLOPs are per inference; parameters and stored features are in millions.

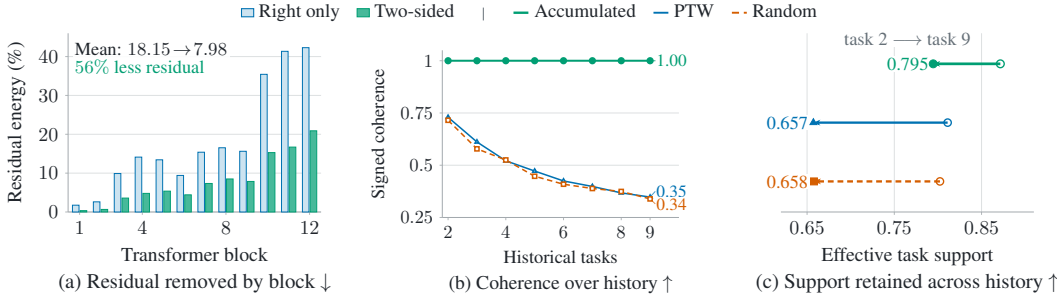
Method	GFLOPs	Params. (M)	Stored (M)	$A_T \uparrow$	$\bar{A} \uparrow$
L2P (Wang et al., 2022b)	70.14	0.480	0	71.26	76.13
DualPrompt (Wang et al., 2022a)	70.26	0.060	0	68.22	73.81
CODA-Prompt (Smith et al., 2023)	70.61	0.380	0	74.05	78.14
InfLoRA (Liang & Li, 2024)	35.12	0.369	0.10	74.75	80.67
SD-LoRA (Wu et al., 2025)	35.12	0.369	0	77.34	82.04
TIP-LoRA	35.12	0.442	0	**79.61**	**84.62**

Table 4: SVD-threshold sensitivity on ten-task ImageNet-R (seed 1993); ranks average over Q/V maps before task 10.

τ	$\bar{k}_{\text{PTW}}$	$\bar{k}_{\text{hist}}$	$A_T \uparrow$	$\bar{A} \uparrow$	$F_T \downarrow$
0.90	287.9	55.3	79.45	84.98	5.37
0.93	327.1	68.0	79.77	85.03	5.14
0.95	361.7	78.0	79.62	85.18	5.68
0.97	410.4	89.4	79.53	85.33	6.14
0.99	498.7	101.7	79.80	85.35	6.50

5.3 Method Analysis

Following Section 4, we therefore test three points in sequence: whether the branches cooperate, whether two-sided control removes interference left by an input-side constraint, and whether cumulative updates retain the paired directions needed for that control.



(a) Residual removed by block ↓ (b) Coherence over history ↑ (c) Support retained across history ↑

Figure 4: Stability diagnostics on CUB-200. (a) Residual old-task gradient energy. (b) Signed coherence across task boundaries. (c) Effective task support; filled markers denote task 9.

Branch cooperation. On CUB-200, the combined model improves final/average accuracy by 1.31/1.04 points over plasticity alone and 1.57/1.25 points over stability alone (Figure 3a). Equal-total-rank CIFAR-100 controls retain gains of 0.45/0.33 and 0.79/0.70 points, respectively (panel b). Adding stability also reduces CUB-200 forgetting by 1.50 points (panel c).

Output-side control. Having established branch cooperation, we isolate the stability projection. Before CUB-200 task 10, disjoint sets of 32 old-task images construct rank-matched input/response bases and measure residual old-task gradient energy. Figure 4(a) shows a reduction in every Transformer block: averaged over blocks and query/value maps, the residual falls from 18.15% with input-side projection to 7.98% with two-sided projection. This 56.0% reduction supports the output-side control predicted by Proposition 1.

Historical directions. The preceding control uses old signals, whereas TIP-LoRA must estimate their directions without retaining them. We therefore evaluate the cumulative update. For singular pair k, task τ contributes $c_{\tau k} = \mathbf{u}_k^\top \Delta \mathbf{W}_\tau \mathbf{v}_k$; the ratio $|\sum_\tau c_{\tau k}| / \sum_\tau |c_{\tau k}|$ measures whether these contributions reinforce, while effective task support rules out domination by one task. In Figure 4(b)–(c), the cumulative basis stays near 1.000 coherence and at least 0.795 support. Before task 10, the PTW and random bases reach only 0.348 and 0.339 coherence, with about 0.657 support. The cumulative SVD retains shared, coherent directions, providing evidence for Assumption 2.

6 Conclusion

TIP-LoRA controls low-rank updates through two-sided interference-aware projection to reduce interference in exemplar-free CIL. A Taylor analysis connects local old-task loss changes to layer inputs and backpropagated gradients. The method explores within a fixed PTW subspace for plasticity and uses the cumulative LoRA update to refresh the stability projectors as tasks arrive. Across five benchmarks, TIP-LoRA achieves competitive accuracy without storing old-task examples. Branch ablations, old-task loss-gradient measurements, and cross-task update coherence support the two-sided projection and cumulative stability basis.

THE USE OF LARGE LANGUAGE MODELS

Generative AI tools assisted with language and structural editing, literature discovery and reference formatting, editing analysis and figure-generation code, and improving the presentation of figures and tables. They also provided feedback on the presentation of the experimental design and its connection to the theoretical claims. The authors reviewed and revised all AI-assisted text and code, verified cited works against the original sources, and checked reported values and visualizations against the underlying experimental outputs. The authors take responsibility for the final text, code, figures, claims, and conclusions.

REFERENCES

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. URL https://arxiv.org/abs/2104.14294.

Satrajit Chatterjee. Coherent gradients: An approach to understanding generalization in gradient descent-based optimization. In *International Conference on Learning Representations*, 2020.

Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 556–572, 2018.

Qiankun Gao, Chen Zhao, Yifan Sun, Teng Xi, Gang Zhang, Bernard Ghanem, and Jian Zhang. A unified continual learning framework with general parameter-efficient tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11483–11493, 2023.

Lingfeng He, De Cheng, Huaijie Wang, Xi Yang, Nannan Wang, and Xinbo Gao. Task-driven subspace decomposition for knowledge sharing and isolation in lora-based continual learning. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2603.00191.

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

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15262–15271, 2021b.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Borui Kang, Lei Wang, Zhiping Wu, Tao Feng, Yawen Li, Yang Gao, and Wenbin Li. Dynamic multi-layer null space projection for vision-language continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2077–2086, 2025.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

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

Muhammad Rifki Kurniawan, Xiang Song, Zhiheng Ma, Yuhang He, Yihong Gong, Yang Qi, and Xing Wei. Evolving parameterized prompt memory for continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 13301–13309, 2024.

Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information Fusion*, 58:52–68, 2020. doi: 10.1016/j.inffus.2019.12.004.

Kai Li, Jiafeng Li, Lianghua He, and Ying Wen. DGS: Dual gradient and semantic-shift guided low-rank adaptation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32267–32277, 2026.

Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision*, pp. 614–629. Springer, 2016.

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 (CVPR)*, pp. 23638–23647, 2024.

Xuan Liu and Xiaobin Chang. LoRA subtraction for drift-resistant space in exemplar-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Arun Mallya and Svetlana Lazebnik. PackNet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019. doi: 10.1016/j.neunet.2019.01.012.

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. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.22389.

Haomiao Qiu, Miao Zhang, Ziyue Qiao, Weili Guan, Min Zhang, and Liqiang Nie. SplitLoRA: Balancing stability and plasticity in continual learning through gradient space splitting. In *International Conference on Learning Representations*, 2026.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021.

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 (CVPR)*, pp. 11909–11919, 2023.

Yuwen Tan, Qinhao Zhou, Xiang Xiang, Ke Wang, Yuchuan Wu, and Yongbin Li. Semantically-shifted incremental adapter-tuning is a continual vitransformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23252–23262, 2024. URL https://arxiv.org/abs/2403.19979v1.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset.

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*, 2023.

Xiequn Wang, Zhan Zhuang, and Yu Zhang. PLAN: Proactive low-rank allocation for continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2909–2918, 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 (ECCV)*, 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 (CVPR)*, pp. 139–149, 2022b.

Xinyao Wu, Zhe Xu, and Raymond Kai-Yu Tong. Continual learning in medical image analysis: A survey. *Computers in Biology and Medicine*, 182:109206, 2024. doi: 10.1016/j.compbiomed.2024.109206.

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 *International Conference on Learning Representations (ICLR)*, 2025.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, volume 70, pp. 3987–3995. PMLR, 2017.

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

Hao Zhu, Yifei Zhang, Junhao Dong, and Piotr Koniusz. BiLoRA: Almost-orthogonal parameter spaces for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Piotr Zielinski, Shankar Krishnan, and Satrajit Chatterjee. Weak and strong gradient directions: Explaining memorization, generalization, and hardness of examples at scale. *arXiv preprint arXiv:2003.07422*, 2020. URL `https://arxiv.org/abs/2003.07422`.

APPENDIX CONTENTS

A Supporting Derivations 13
Local Loss Expansion 13
From Historical Gradients to Cumulative Task Updates 13
Conditional Singular-Subspace Recovery 14
Terms Removed by the Two Projections 15
Weights Shared Across Tokens 15
B Implementation and Evaluation Details 15
Benchmark and Ablation Protocols 15
Backbone and Projected LoRA Branches 16
Task-Wise Comparisons 16
Branch Analysis 16
Sensitivity to the SVD Energy Threshold 18
C Finite-Perturbation Check 17
D Claim Diagnostics 17
Input- and Output-Side Interference Control 17
Direction Coherence in Cumulative Task Updates 18

A SUPPORTING DERIVATIONS

A.1 LOCAL LOSS EXPANSION

For this proof, let $\ell_s(\mathbf{z})$ denote the old-task loss obtained when the current layer output is set to $\mathbf{z}$, with the downstream network and all other quantities fixed. Set $\mathbf{d}_s = \Delta\mathbf{W}\mathbf{h}_s$, the layer-output change. By convexity, the segment $\mathbf{z}_s + t\mathbf{d}_s$, $0 \le t \le 1$, lies in the neighborhood from Assumption 1. The input, other weights, and stochastic masks are fixed. Define the scalar path $\phi_s(t) = \ell_s(\mathbf{z}_s + t\mathbf{d}_s)$. By the chain rule,

$$\phi_s'(t) = \nabla_{\mathbf{z}}\ell_s(\mathbf{z}_s + t\mathbf{d}_s)^\top \mathbf{d}_s. \tag{18}$$

The fundamental theorem of calculus therefore gives

$$\begin{aligned} \delta\ell_s(\Delta\mathbf{W}) &= \phi_s(1) - \phi_s(0) \\ &= \int_0^1 \nabla_{\mathbf{z}}\ell_s(\mathbf{z}_s + t\mathbf{d}_s)^\top \mathbf{d}_s \, dt \\ &= \nabla_{\mathbf{z}}\ell_s(\mathbf{z}_s)^\top \mathbf{d}_s + R_s \\ &= \mathbf{e}_s^\top \Delta\mathbf{W}\mathbf{h}_s + R_s, \end{aligned} \tag{19}$$

where

$$R_s = \int_0^1 [\nabla_{\mathbf{z}}\ell_s(\mathbf{z}_s + t\mathbf{d}_s) - \nabla_{\mathbf{z}}\ell_s(\mathbf{z}_s)]^\top \mathbf{d}_s \, dt, \tag{20}$$

where t parameterizes the segment. Since the gradient is L_s-Lipschitz,

$$|R_s| \le \int_0^1 L_s t \|\mathbf{d}_s\|_2^2 \, dt = \frac{L_s}{2}\|\mathbf{d}_s\|_2^2 \le \frac{L_s}{2}\|\Delta\mathbf{W}\|_2^2 \|\mathbf{h}_s\|_2^2. \tag{21}$$

Equation (19) together with this bound is exactly Eq. (4). For fixed s and reference network, $L_s\|\mathbf{h}_s\|_2^2/2$ is independent of the perturbation within this neighborhood. This proves the stated local order as $\Delta\mathbf{W} \to 0$. The quadratic bound uses Lipschitz continuity of the gradient; twice continuous differentiability is a sufficient local condition. For piecewise-smooth networks, the argument applies to neighborhoods satisfying Assumption 1.

A.2 FROM HISTORICAL GRADIENTS TO CUMULATIVE TASK UPDATES

For a single linear occurrence, $d\ell_s = \mathbf{e}_s^\top (d\mathbf{W})\mathbf{h}_s$, so $\nabla_{\mathbf{W}}\ell_s = \mathbf{e}_s\mathbf{h}_s^\top$. The decomposition in Eq. (10) therefore gives, for each historical gradient contribution q,

$$\begin{aligned} \mathbf{e}_q\mathbf{h}_q^\top = {} & \mathbf{U}_\star \mathbf{a}_q \mathbf{b}_q^\top \mathbf{V}_\star^\top + \mathbf{U}_\star \mathbf{a}_q (\mathbf{r}_q^h)^\top \\ & + \mathbf{r}_q^e \mathbf{b}_q^\top \mathbf{V}_\star^\top + \mathbf{r}_q^e (\mathbf{r}_q^h)^\top. \end{aligned} \tag{22}$$

Thus the signed first-order accumulation $-\sum_{q\in\mathcal{Q}_{<t}} \eta_q \mathbf{e}_q \mathbf{h}_q^\top$ contains the component supported jointly on the loss-response and input subspaces, $\mathbf{U}_\star \mathbf{C}_{t-1}^{\mathrm{grad}} \mathbf{V}_\star^\top$, where

$$\mathbf{C}_{t-1}^{\mathrm{grad}} = -\sum_{q\in\mathcal{Q}_{<t}} \eta_q \mathbf{a}_q \mathbf{b}_q^\top. \tag{23}$$

This cross-statistic explains why accumulated weight updates can retain both loss-response and input directions. The second equality in Eq. (9) defines $\mathbf{R}_{t-1}^{\mathrm{opt}}$ exactly as the discrepancy between this raw gradient accumulation and the cumulative LoRA update matrix. Let

$$\mathbf{R}_{t-1}^{\mathrm{rem}} = \mathbf{R}_{t-1}^{\mathrm{opt}} - \mathbf{U}_\star(\mathbf{U}_\star^\top \mathbf{R}_{t-1}^{\mathrm{opt}} \mathbf{V}_\star)\mathbf{V}_\star^\top. \tag{24}$$

Combining Eq. (22) with Eq. (9) yields

$$\begin{aligned} \mathbf{C}_{t-1} &= \mathbf{C}_{t-1}^{\mathrm{grad}} + \mathbf{U}_\star^\top \mathbf{R}_{t-1}^{\mathrm{opt}} \mathbf{V}_\star, \\ \mathbf{E}_{t-1} &= \mathbf{R}_{t-1}^{\mathrm{rem}} - \sum_{q\in\mathcal{Q}_{<t}} \eta_q \big[\mathbf{U}_\star \mathbf{a}_q (\mathbf{r}_q^h)^\top + \mathbf{r}_q^e \mathbf{b}_q^\top \mathbf{V}_\star^\top + \mathbf{r}_q^e (\mathbf{r}_q^h)^\top\big]. \end{aligned} \tag{25}$$

Consequently,

$$\begin{aligned} \|\mathbf{E}_{t-1}\|_2 \le \|\mathbf{R}_{t-1}^{\mathrm{rem}}\|_2 \\ + \sum_{q\in\mathcal{Q}_{<t}} |\eta_q| \big(\|\mathbf{a}_q\|_2 \|\mathbf{r}_q^h\|_2 + \|\mathbf{r}_q^e\|_2 \|\mathbf{b}_q\|_2 + \|\mathbf{r}_q^e\|_2 \|\mathbf{r}_q^h\|_2\big). \end{aligned} \tag{26}$$

Thus, the residual bound supplies a sufficient condition for Assumption 2: its right-hand side must be smaller than half the weakest retained singular value.

A.3 Conditional Singular-Subspace Recovery

The rank-k component $\mathbf{U}_\star \mathbf{C}_{t-1} \mathbf{V}_\star^\top$ has the same nonzero singular values as $\mathbf{C}_{t-1}$. Assumption 2 defines γ_{t-1} and ε_{t-1} and gives $\|\mathbf{E}_{t-1}\|_2 \le \varepsilon_{t-1} < \gamma_{t-1}/2$. The singular-value perturbation bound gives $\sigma_k(\mathbf{M}_{t-1}) \ge \gamma_{t-1} - \varepsilon_{t-1} > 0$, while $\sigma_{k+1}(\mathbf{M}_{t-1}) \le \varepsilon_{t-1}$ when the latter singular value exists. Consequently,

$$\sigma_k(\mathbf{M}_{t-1}) - \sigma_{k+1}(\mathbf{M}_{t-1}) \ge \gamma_{t-1} - 2\varepsilon_{t-1} > 0, \tag{27}$$

where the final inequality is exactly the condition $\varepsilon_{t-1} < \gamma_{t-1}/2$. Thus Assumption 2 gives a positive spectral gap at rank k. Let $\mathbf{\Sigma}_k$ be the diagonal matrix of retained singular values. From $\mathbf{M}_{t-1}\mathbf{V} = \mathbf{U}\mathbf{\Sigma}_k$,

$$(\mathbf{I} - \mathbf{U}_\star \mathbf{U}_\star^\top)\mathbf{U}\mathbf{\Sigma}_k = (\mathbf{I} - \mathbf{U}_\star \mathbf{U}_\star^\top)\mathbf{E}_{t-1}\mathbf{V}. \tag{28}$$

Multiplying by $\mathbf{\Sigma}_k^{-1}$ and taking norms yields

$$\|(\mathbf{I} - \mathbf{U}_\star \mathbf{U}_\star^\top)\mathbf{U}\|_2 \le \frac{\varepsilon_{t-1}}{\gamma_{t-1} - \varepsilon_{t-1}}. \tag{29}$$

For equal-dimensional subspaces the largest sine of the principal angles is symmetric in their order. The left-hand side therefore equals $\|\mathbf{P}_U^\perp \mathbf{U}_\star\|_2$. Applying the argument to $\mathbf{M}_{t-1}^\top$ gives the corresponding right-side bound:

$$\max\{\|\mathbf{P}_U^\perp \mathbf{U}_\star\|_2, \|\mathbf{P}_V^\perp \mathbf{V}_\star\|_2\} \le \frac{\varepsilon_{t-1}}{\gamma_{t-1} - \varepsilon_{t-1}}. \tag{30}$$

This is Eq. (13). Combining it with Eq. (10) also yields

$$\begin{aligned} \|\mathbf{P}_U^\perp \mathbf{e}_s\|_2 &\le \frac{\varepsilon_{t-1}}{\gamma_{t-1} - \varepsilon_{t-1}} \|\mathbf{a}_s\|_2 + \|\mathbf{r}_s^e\|_2, \\ \|\mathbf{P}_V^\perp \mathbf{h}_s\|_2 &\le \frac{\varepsilon_{t-1}}{\gamma_{t-1} - \varepsilon_{t-1}} \|\mathbf{b}_s\|_2 + \|\mathbf{r}_s^h\|_2. \end{aligned} \tag{31}$$

These are the two residuals entering Proposition 1. Thus, when the retained component dominates the residual, the leading left and right singular subspaces of $\mathbf{M}_{t-1}$ approximate the retained loss-response and input subspaces used to construct the stability projectors.

A.4 Terms Removed by the Two Projections

Let $\mathbf{P}_U$ and $\mathbf{P}_V$ be the projectors fixed in the main text. Write $\mathbf{e}_s = \mathbf{P}_U\mathbf{e}_s + \bar{\mathbf{e}}_s$ and $\mathbf{h}_s = \mathbf{P}_V\mathbf{h}_s + \bar{\mathbf{h}}_s$. Expanding the first-order term for the two-sided update gives

$$\begin{aligned}\mathbf{e}_s^\top \Delta\mathbf{W}_\perp \mathbf{h}_s = &\underbrace{(\mathbf{P}_U\mathbf{e}_s)^\top \Delta\mathbf{W}_\perp(\mathbf{P}_V\mathbf{h}_s)}_{\text{0: either constraint}} \\ &+ \underbrace{(\mathbf{P}_U\mathbf{e}_s)^\top \Delta\mathbf{W}_\perp \bar{\mathbf{h}}_s}_{\text{0: left constraint}} \\ &+ \underbrace{\bar{\mathbf{e}}_s^\top \Delta\mathbf{W}_\perp(\mathbf{P}_V\mathbf{h}_s)}_{\text{0: right constraint}} + \bar{\mathbf{e}}_s^\top \Delta\mathbf{W}\bar{\mathbf{h}}_s.\end{aligned} \tag{32}$$

The zeros follow from Eq. (5) and symmetry of the projectors. This proves Eq. (6); Cauchy–Schwarz applied to the remaining term proves Eq. (7).

Under Assumption 1, the finite-update terms satisfy

$$|R_s| \le \frac{L_s}{2}\|\Delta\mathbf{W}\|_2^2\|\bar{\mathbf{h}}_s\|_2^2, \tag{33}$$

and consequently

$$|\delta\ell_s(\Delta\mathbf{W}_\perp)| \le \|\Delta\mathbf{W}\|_2\|\bar{\mathbf{e}}_s\|_2\|\bar{\mathbf{h}}_s\|_2 + \frac{L_s}{2}\|\Delta\mathbf{W}\|_2^2\|\bar{\mathbf{h}}_s\|_2^2. \tag{34}$$

To obtain these bounds, observe that the output change is $\Delta\mathbf{W}_\perp\mathbf{h}_s = \mathbf{P}_U^\perp\Delta\mathbf{W}\bar{\mathbf{h}}_s$, whose norm is at most $\|\Delta\mathbf{W}\|_2\|\bar{\mathbf{h}}_s\|_2$. Substitution into Eq. (4) proves Eq. (33). Combining the first-order and remainder bounds gives Eq. (34). The gradient residual controls first-order coupling, whereas the quadratic remainder is bounded through the layer-output change. When $\bar{\mathbf{e}}_s = 0$, this remainder still governs the local loss change.

The token-shared extension below uses the same projectors and residual notation in matrix form.

A.5 Weights Shared Across Tokens

For a sample with n_s token occurrences, let $\mathbf{H}_s \in \mathbb{R}^{d_{\text{in}}\times n_s}$ collect their input columns and $\mathbf{Y}_s = \mathbf{W}\mathbf{H}_s \in \mathbb{R}^{d_{\text{out}}\times n_s}$ their outputs. The downstream sample loss is a function of the whole matrix; write $\mathbf{D}_s = \nabla_{\mathbf{Y}_s}\ell_s$ for its output gradient. For token index $j \in \{1,\dots,n_s\}$, let $\mathbf{h}_{s,j}$ and $\mathbf{e}_{s,j}$ be column j of $\mathbf{H}_s$ and $\mathbf{D}_s$. Then $\nabla_{\mathbf{W}}\ell_s = \sum_{j=1}^{n_s}\mathbf{e}_{s,j}\mathbf{h}_{s,j}^\top$. Each $\mathbf{e}_{s,j}$ is a derivative of the whole sample loss, so attention-mediated interactions are included. With tr denoting trace,

$$\nabla_{\mathbf{W}}\ell_s = \mathbf{D}_s\mathbf{H}_s^\top, \qquad \delta\ell_s = \mathrm{tr}(\mathbf{D}_s^\top\Delta\mathbf{W}\mathbf{H}_s) + R_s. \tag{35}$$

Define $\bar{\mathbf{D}}_s = \mathbf{P}_U^\perp\mathbf{D}_s$ and $\bar{\mathbf{H}}_s = \mathbf{P}_V^\perp\mathbf{H}_s$. For the stability update, the same algebra and Frobenius Cauchy–Schwarz give

$$\begin{aligned}\left|\mathrm{tr}(\mathbf{D}_s^\top\Delta\mathbf{W}_\perp\mathbf{H}_s)\right| &= \left|\mathrm{tr}(\bar{\mathbf{D}}_s^\top\Delta\mathbf{W}\bar{\mathbf{H}}_s)\right| \\ &\le \|\Delta\mathbf{W}\|_2\|\bar{\mathbf{D}}_s\|_F\|\bar{\mathbf{H}}_s\|_F.\end{aligned} \tag{36}$$

Under the matrix version of Assumption 1, using the Frobenius norm, the remainder satisfies $|R_s| \le (L_s/2)\|\Delta\mathbf{W}\|_2^2\|\bar{\mathbf{H}}_s\|_F^2$. Thus the same two-residual mechanism applies to a token-shared weight, with the other layers and the input matrix held fixed.

B Implementation and Evaluation Details

B.1 Benchmark and Ablation Protocols

Tables 1–2 report TIP-LoRA with the projected LoRA branches and current-task cross-entropy. The benchmark record specifies a ViT-B/16 backbone, SGD, batch size 48, the normalized cosine classifier described in Section 4.3, and the task schedules shown in the tables. The implementation applies cross-entropy to scaled, normalized cosine logits; the margin is zero in all reported configurations.

Let f_t denote the network with the projected LoRA updates and let c_t be the cosine head for classes in $\mathcal{C}_t$. The current-task objective is

$$\mathcal{L}_t = \frac{1}{n_t} \sum_{(x,y)\in\mathcal{D}_t} \ell_{\mathrm{CE}}\big(c_t(f_t(x)), y\big). \tag{37}$$

The reported TIP-LoRA configuration uses plasticity/stability ranks of 10/2 on CIFAR-100, ImageNet-R with 5 or 10 tasks, and ImageNet-A; 8/4 on CUB-200; 6/4 on DomainNet; and 2/8 on ImageNet-R with 20 tasks. Training uses 20 epochs per task with learning rate 0.01 on CIFAR-100, five-task ImageNet-R, and ImageNet-A; 0.007 on ten- and twenty-task ImageNet-R; and 0.005 on CUB-200. DomainNet uses 5 epochs and learning rate 0.006.

Metrics. Following standard CIL evaluation (Gao et al., 2023; Liang & Li, 2024; Wu et al., 2025), let A_t be the accuracy over all seen classes after task t. We report final accuracy A_T and average incremental accuracy $\bar{A} = T^{-1}\sum_{t=1}^{T} A_t$. For the branch ablation, if $a_{t,i}$ is the accuracy on task i after learning task t, final forgetting is (Chaudhry et al., 2018)

$$F_T = \frac{1}{T-1}\sum_{i=1}^{T-1}\left(\max_{i\le t<T} a_{t,i} - a_{T,i}\right). \tag{38}$$

Higher is better for A_T and $\bar{A}$, whereas lower is better for F_T. All reported accuracies are percentages.

Result sources and uncertainty. Baseline entries retain the means and uncertainty conventions reported by their cited sources. TIP-LoRA reports the mean and sample standard deviation over seeds 1993, 1994, and 1995. On CIFAR-100, the three runs give final accuracies of 89.02, 88.54, and 87.77 and average accuracies of 93.302, 92.765, and 92.395, yielding 88.44 ± 0.63 and 92.82 ± 0.46.

B.2 Backbone and Projected LoRA Branches

The architecture study uses ViT-B/16 with 12 blocks, a feature dimension of 768, and 12 attention heads. Query and value weights are adapted; the remaining pre-trained parameters are frozen. At task t, the forward pass uses LoRA modules introduced through task t.

B.3 Task-Wise Comparisons

The baseline trajectories in Figure 2 were digitized from the published vector figures using their original axis scales and rounded to 0.01 percentage points. In the supervised panels, the TIP-LoRA trajectories are the task-wise means over seeds 1993, 1994, and 1995. They end at 57.56% on ten-task ImageNet-A and 79.61% on ten-task ImageNet-R. The DINO panel uses the seed-1993 run and ends at 71.03%; the corresponding multi-seed experiment is not yet available.

B.4 Branch Analysis

Figure 3 uses seed 1993 throughout. The CUB-200 comparison trains the 8/4 dual-branch model and its 8/0 and 0/4 branch-removal controls for 20 epochs per task with learning rate 0.005 and batch size 32. All three runs use the same class order and current-task-only optimization.

The CIFAR-100 comparison controls total branch rank. It compares the 10/2 dual-branch model with 12/0 and 0/12 single-branch models under the same class order, 20-epoch schedule, learning rate 0.01, and batch size 48. Final accuracy is the last all-seen-class accuracy, average accuracy is the mean of the ten task-boundary accuracies, and forgetting is recomputed from the complete task accuracy matrix using Section 5.1.

B.5 Sensitivity to the SVD Energy Threshold

We vary the cumulative-SVD energy threshold used to select k on ten-task ImageNet-R. All five runs use seed 1993 and the same training configuration. Table 7 shows that final and average incremental accuracy each vary by less than 0.4 percentage points across thresholds 0.90–0.99.

Table 5: Accuracy (%) under different SVD energy thresholds on ten-task ImageNet-R (seed 1993).

Energy threshold	$A_T \uparrow$	$\bar{A} \uparrow$
0.90	79.45	84.98
0.93	79.77	85.03
0.95	79.62	85.18
0.97	79.53	85.33
0.99	79.80	85.35

C Finite-Perturbation Check

This experiment checks the local expansion in Eq. (4) at the checkpoint before task 10 of an ImageNet-R run. The test uses 24 held-out old-task examples. In each block, the saved task-10 projected query and value contributions are injected while the remaining weights are fixed. We compare the resulting class-token output change with its automatic-differentiation first-order prediction. The joint update norm is normalized by that of the corresponding PTWs. Multipliers are $1/8, 1/4, 1/2, 1, 2$, and the log–log remainder fit uses the four values up to one.

Table 6: Finite-perturbation remainder by Transformer block. Update norms are relative to the corresponding PTW (%).

Block	Slope	Update norm	Block	Slope	Update norm
1	1.773	2.68	7	1.994	4.00
2	1.659	2.87	8	1.994	4.01
3	1.504	3.36	9	1.990	4.07
4	1.730	3.71	10	1.991	4.08
5	1.975	3.89	11	1.987	4.07
6	1.982	3.91	12	1.992	3.84

Remainder slopes range from 1.50 to 1.99 (1.975–1.994 in blocks 5–12), showing layer-dependent approximation quality.

C.1 Efficiency comparison

Computation, parameters, and storage. On ten-task ImageNet-R, TIP-LoRA requires 35.12 billion floating-point operations (GFLOPs) per inference, matching InfLoRA and SD-LoRA and half the 70.14–70.61 GFLOPs reported for three prompt-based baselines. It adds 0.442M LoRA parameters per task (versus 0.369M for the two LoRA baselines) and stores no old-task features.

D Claim Diagnostics

D.1 Input- and Output-Side Interference Control

The first diagnostic tests the interference mechanism independently of the construction of the stability projectors from cumulative task updates. For an old-task image s, its weight gradient is

$$\mathbf{G}_s = \sum_j \mathbf{e}_{s,j} \mathbf{h}_{s,j}^\top, \tag{39}$$

where $\mathbf{h}_{s,j}$ is the input to the selected linear map and $\mathbf{e}_{s,j}$ is the loss response propagated to its output. We use these two signals to construct rank-matched input- and output-side bases. Their complementary projectors are denoted by $\mathbf{Q}^{\perp}$ and $\mathbf{P}^{\perp}$, respectively. The residual energy of a projected gradient is

$$\rho(\mathbf{P}, \mathbf{Q}) = \frac{\sum_s \|\mathbf{P}^{\perp}\mathbf{G}_s\mathbf{Q}^{\perp}\|_F^2}{\sum_s \|\mathbf{G}_s\|_F^2}. \tag{40}$$

We evaluate a CUB-200 checkpoint immediately before task 10. A fixed set of 32 old-task test images constructs the two bases, while a disjoint set of 32 images measures Eq. (40). The checkpoint, images, ranks, and right projector remain unchanged between conditions; only the output-side projector is added. Under this control, the residual falls from 18.15% with right-only projection to 7.98% with two-sided projection, a relative reduction of 56.0%. Thus, the left projector removes interference associated with the loss response that is invisible to the input-side constraint alone. Old-task data are used only for this diagnostic and are never used by TIP-LoRA during continual training.

D.2 Direction Coherence in Cumulative Task Updates

The second diagnostic asks whether cumulative task updates retain consistent left and right directions for the stability projectors. Let $\Delta\mathbf{W}_\tau$ be the update produced by task τ, define $\mathbf{M}_{t-1} = \sum_{\tau=1}^{t-1} \Delta\mathbf{W}_\tau$, and let $(\mathbf{u}_k, \mathbf{v}_k)$ be a left–right singular-vector pair of $\mathbf{M}_{t-1}$. The signed contribution of task τ to this pair is

$$c_{\tau k} = \mathbf{u}_k^{\top} \Delta\mathbf{W}_\tau \mathbf{v}_k. \tag{41}$$

We summarize cancellation with signed coherence and distinguish repeated support from a single-task spike with effective task support:

$$C_k = \frac{|\sum_\tau c_{\tau k}|}{\sum_\tau |c_{\tau k}|}, \qquad S_k = \frac{(\sum_\tau |c_{\tau k}|)^2}{n \sum_\tau c_{\tau k}^2}, \tag{42}$$

where n is the number of completed tasks. A large C_k means that task-wise contributions reinforce rather than cancel, whereas a large S_k means that the direction is supported across several tasks.

We compute both quantities before CUB-200 task 10 across all 24 adapted linear maps and 2,443 retained modes. The singular directions of the cumulative task-update matrix obtain mean signed coherence 1.000 and effective task support 0.795. Using PTW or rank-matched random singular pairs instead yields coherence 0.348 and 0.339, with support 0.657 and 0.658. The cumulative task-update directions are therefore both consistently oriented and broadly supported across tasks. Together, these results support using the leading left and right singular directions of the cumulative task-update matrix to construct the stability projectors without retaining old examples.

D.3 Sensitivity to Rank Selection

On ten-task ImageNet-R (seed 1993), increasing the SVD energy threshold from 0.90 to 0.99 markedly raises the retained PTW and cumulative-update ranks, yet final and average incremental accuracies each vary by less than 0.4 points while forgetting ranges from 5.14 to 6.50 points (Table 7).

Table 7: SVD energy-threshold sensitivity on ten-task ImageNet-R (seed 1993); ranks are averaged over query and value matrices in all 12 blocks before the final task.

Threshold τ	$\bar{k}_{\mathrm{PTW}}$	$\bar{k}_{\mathrm{hist}}$	$A_T \uparrow$	$\bar{A} \uparrow$	$F_T \downarrow$
0.90	287.9	55.3	79.45	84.98	5.37
0.93	327.1	68.0	79.77	85.03	5.14
0.95	361.7	78.0	79.62	85.18	5.68
0.97	410.4	89.4	79.53	85.33	6.14
0.99	498.7	101.7	79.80	85.35	6.50