CHAINLORA: GEOMETRY-PRESERVING TASK VECTOR MERGING FOR CONTINUAL LEARNING IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual parameter-efficient fine-tuning for large language models (LLMs) must balance retention of previously acquired knowledge, adaptation to new tasks, and strict parameter budgets. We present **ChainLoRA**, a replay-free continual merging framework built on chain-updated task-vector geometry. From a parameter-merging perspective, we formulate a geometric view of forgetting through a measurable interaction between task updates, separating directional overlap from coefficient coupling. Building on this view, ChainLoRA combines chain-updated training with post-stream adaptive SVD merging. During training, initialization and a one-sided orthogonality proxy use only the last carrier, keeping their historical-state footprint and regularization overhead constant as the task stream grows. At merging time, Adaptive SVD extracts a shared carrier and aligns it to the latest task through Procrustes adaptation. Our theoretical analysis shows that Procrustes adaptation facilitates geometric approximate separation of shared and task-specific components. The one-sided proxy further bounds inter-task interference. An effective-rank penalty additionally promotes efficient utilization of the task subspace during continual learning. Experiments show that ChainLoRA achieves state-of-the-art performance among the evaluated replay-free methods on the Large and SuperNI benchmarks, while remaining competitive on Standard CL and attaining almost the closest average scores to the evaluated replay-based method across all three benchmarks.

1 INTRODUCTION

Foundation models, e.g. large language models (LLMs), have shown remarkable generalization and reasoning abilities across diverse domains. When they are subsequently finetuned on a stream of domains (Karanikolas et al., 2023), later updates often overwrite earlier ones—*catastrophic forgetting* (McCloskey & Cohen, 1989; Liu et al., 2024). Continual learning (Shi et al., 2025) studies how to acquire new knowledge without this collapse. *Regularization methods* (Kirkpatrick et al., 2017) keep parameters close to a previous solution, but can hinder adaptation to new tasks. *Replay methods* (Rebuffi et al., 2017) revisit old data, which is often unavailable or costly during post-training. *Architectural isolation* (Tong et al., 2025) assigns tasks to separate modules, at the price of extra parameters and weaker sharing (Verma et al., 2021), which may further limit generalization across tasks. These limits raise a concrete question: *Can a CL method avoid catastrophic forgetting without compromising cross-task generalization or relying on data replay?*

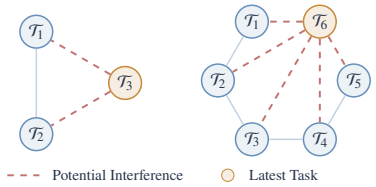

Figure 1: Potential task interference in short CL (left) and long CL (right).

Setting replay-based approaches aside, we focus on *replay-free* continual adaptation over *long* task streams. As tasks accumulate, maintaining previously acquired knowledge while accommodating new tasks becomes increasingly challenging under a limited parameter budget. We consider two representative strategies: orthogonality-based adaptation, exemplified by O-LoRA (Wang et al., 2023), and architectural isolation through task-specific modules and routing. These strategies motivate two closely related challenges.

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.

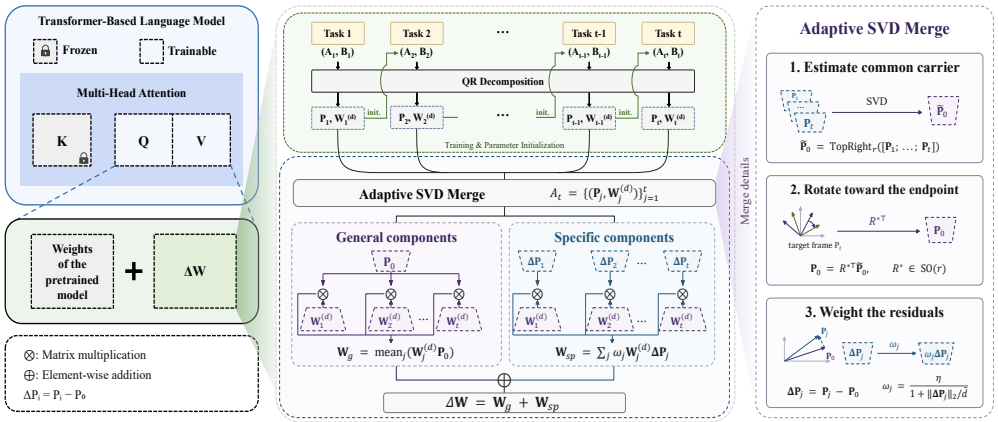

Figure 2: Overview of ChainLoRA. *During* the stream, each update is split by thin QR into a row-orthonormal carrier $\mathbf{P}_i$ and magnitude $\mathbf{W}_i^{(d)}$; the next task is initialized from the last carrier, and a one-sided proxy loss encourages disentanglement of common knowledge from that of the task-specific knowledge. *After* the stream, Adaptive SVD estimates a shared rank-r carrier, rotates it in $\mathrm{SO}(r)$ — the group of r-dimensional rotations — onto the latest-task carrier; residuals are then weighted by their distance from that, as described in Appendix D.3.

The first challenge is *growing inter-task interference*. As illustrated in Fig. 1, each incoming task may interact with an increasing number of historical tasks, creating more opportunities for conflicting updates. OLoRA (Wang et al., 2023) mitigates such interference by separating task-update directions. However, orthogonality only provides an optimization objective without explicitly addressing how multiple task updates should be merged. Moreover, constraining each new update against previous ones incurs growing training costs. Architectural isolation instead assigns tasks to different modules. Inter-task interference may persist when multiple tasks are routed to shared components with limited parameter capacity.

The second challenge is *efficient use of limited adaptation capacity* as Fig. 3. OLoRA enforcing orthogonality between entire task updates can be overly restrictive, as different tasks often share transferable knowledge. In the extreme, even splitting the same task into two sequential stages would force their updates into separate directions, unnecessarily consuming additional parameter space. Such over-separation reduces the capacity available for long-term adaptation as more tasks arrive. Architectural isolation faces a similar limitation: allocating new parameters to different tasks primarily postpones capacity saturation through model expansion, rather than improving the reuse of existing capacity. Meanwhile, isolating tasks in separate modules may also limit cross-task generalization.

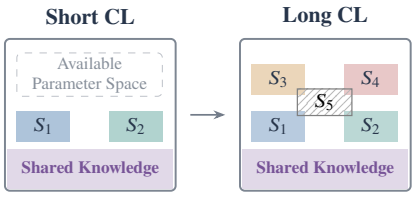

Figure 3: Parameter-space crowding in long-horizon continual learning: short task streams (left) versus long task streams (right). S_5 leads to forggeting.

We treat that question as a problem of *task-vector geometry* rather than of allocating extra modules. Building upon orthogonalization, we further decompose each task update into a shared component and a task-specific component, instead of enforcing orthogonality between entire task vectors. We encourage the task-specific components to be orthogonal to the shared component and to one another, allowing transferable knowledge to occupy shared directions while isolating task-specific variations. This structured decomposition preserves the benefits of orthogonality without unnecessarily separating knowledge that can be reused across tasks.

A well-adopted observation is that a low-dimensional update can match full-parameter fine-tuning (Wang et al., 2023). Write $\mathbf{W} = \mathbf{W}_0 + \Delta\mathbf{W}$ with $\mathbf{W}_0$ frozen and $\Delta\mathbf{W} = \mathbf{BA}$, so CL knowledge lives in a small additive factor. In a continual task stream, that factor is decomposed into tasks and still couples in two ways: the induced directions of different tasks, and their coefficients

can activate those. We specifically standardize each update by a thin QR factorization, taking the row-orthonormal factor as the *task vector* (carrier) and the matching coefficients as magnitude.

The method then acts at two timescales, because the objects available during training and after the stream are not the same. *During* the stream, the post-hoc merge carrier is unavailable: only the last task's carrier can be copied forward. Chain-updated initialization starts the next task from the latest. A one-sided proxy loss encourages disentanglement of common knowledge from that of the task-specific knowledge, while a smoothed group penalty on magnitude columns encourages lower-rank updates, preserving adaptation capacity for future tasks and mitigating potential parameter-space crowding. *After* the stream, Adaptive SVD merging estimates a rank-r shared carrier, aligns to the latest task by Procrustes adaptation, and damps residuals according to their distance from it. Finally, the merged update is absorbed into a single adapter for inference. Highlights of our contribution are:

(1) We cast catastrophic forgetting as an operational geometric quantity in merging: each update indicates a carrier with magnitude, naive merging couples them through inner products, and orthogonality is sufficient for weight-space disentanglement. **(2)** On this geometry, we develop ChainLoRA, combining chain-updated training with a post-stream Adaptive SVD merge. The online penalty and initialization use only the last carrier, so historical cost stays constant in the CL stream. **(3)** **(3)** We provide theoretical analysis showing that Procrustes adaptation facilitates geometric approximate separation of shared and task-specific components, while the one-sided proxy further bounds inter-task interference. **(4)** Experiments indicate strong replay-free performance. On the long-horizon Large stream, ChainLoRA is the strongest among published replay-free LoRA and merging baselines we compare; on shorter Standard CL it approaches replay-augmented methods such as SAPT-LoRA.

2 Related Work

2.1 Continual Learning and Catastrophic Forgetting

Continual learning studies how a model can learn a sequence of tasks while mitigating *catastrophic forgetting*. Existing methods fall into three families. **Regularization-based methods** (Kirkpatrick et al., 2017; Lopez-Paz & Ranzato, 2017; Farajtabar et al., 2020) constrain updates via importance-weighted penalties to preserve past knowledge, but strong constraints can hinder plasticity. **Replay-based methods** (Rebuffi et al., 2017; Rolnick et al., 2019; Buzzega et al., 2020; de Masson D'Autume et al., 2019) mix old or pseudo samples with new data; for foundation-model post-training, storing historical data is often infeasible due to privacy or licensing, and generative replay adds overhead. **Architecture-based (structural isolation) methods** (Aljundi et al., 2017; Mallya & Lazebnik, 2018; Wang et al., 2022b; Verma et al., 2021) allocate a separate module per task, reducing interference at the cost of more parameters and less sharing. Our work keeps the isolation but realizes it geometrically: task share a carrier when they can, and otherwise remain as residuals inside one LoRA budget.

Within LoRA-based continual learning, O-LoRA enforces orthogonality in low-rank updates (Wang et al., 2023), InfLoRA introduces interference-aware subspace control (Liang & Li, 2024), SAPT-LoRA augments replay via synthetic historical samples (Zhao et al., 2024), and recent variants such as LoRM (Salami et al., 2025), CorDA (Yang et al., 2024), and MagMax (Marczak et al., 2024) explore different constraints on low-rank adaptation. They typically optimize per-task adaptation rather than a unified geometry for continual merging.

2.2 Low-Rank Adaptation and Parameter-Efficient Fine-Tuning

Large language models can be adapted with far fewer trainable parameters than the full network. LoRA Hu et al. (2022) parameterizes the update as $\Delta\mathbf{W} = \mathbf{BA}$ (with frozen $\mathbf{W}_0$ and rank $r \ll d$), consistent with the observation that fine-tuning often has low intrinsic dimension Li et al. (2018). These methods reduce cost but typically use a single adapter or LoRA per model (or per task), and do not specify a shared-trunk / residual geometry for composing a sequence of updates.

Parallel to PEFT, recent work studies merging independently trained adapters without original data (Ilharco et al., 2023; Zhang et al., 2023; Yadav et al., 2023; Ortiz-Jimenez et al., 2023; Wortsman et al., 2022), including LoRA-LEGO's structured recombination (Zhao et al., 2025), KnOTS's SVD-oriented task arithmetic (Stoica et al., 2025), and OPCM for streaming updates (Tang et al., 2025).

Most existing formulations are not specifically designed for asymmetric LoRA factor roles under continual adaptation, which motivates our chain-updated geometry-aware merging design.

3 Rethinking Catastrophic Forgetting Through Task Vectors

Inspired by recent findings, e.g., FlyLoRA (Zou et al., 2025), that inter-task orthogonality improves parameter merging, we revisit continual adaptation as the organization of task updates.

3.1 Task Vectors and Interaction Bounds

Given pretrained weights $\mathbf{W}_0$, let the rank-r update induced by task $\mathcal{T}_i$ be $\Delta\mathbf{W}_i \in \mathbb{R}^{n\times m}$. We use a thin QR factorization of the transpose,

$$\Delta\mathbf{W}_i^\top = \mathbf{Q}_i\mathbf{R}_i, \qquad \mathbf{Q}_i \in \mathbb{R}^{m\times r}, \quad \mathbf{R}_i \in \mathbb{R}^{r\times n}, \quad \mathbf{Q}_i^\top\mathbf{Q}_i = \mathbf{I}_r.$$

Definition 1 (Task Vector). *Define $\mathbf{P}_i = \mathbf{Q}_i^\top \in \mathbb{R}^{r\times m}$ and $\mathbf{W}_i^{(d)} = \mathbf{R}_i^\top \in \mathbb{R}^{n\times r}$. Then $\mathbf{P}_i\mathbf{P}_i^\top = \mathbf{I}_r$ and $\Delta\mathbf{W}_i = \mathbf{W}_i^{(d)}\mathbf{P}_i$. We call the row-orthonormal carrier $\mathbf{P}_i$ the task vector of $\mathcal{T}_i$.*

Under the standard QR sign convention (e.g., constraining diagonal entries of $\mathbf{R}_i$ to be positive), this decomposition is unique. Therefore, $\mathbf{P}_i$ is strongly task-dependent and summarizes dominant task-induced variation directions, while $\mathbf{W}_i^{(d)}$ encodes the corresponding magnitudes. The carrier cross-Gram matrix $\mathbf{P}_i\mathbf{P}_j^\top$ measures subspace overlap.

Theorem 1. *For two tasks i, j, let $\Delta\mathbf{W}_i = \mathbf{W}_i^{(d)}\mathbf{P}_i$ and $\Delta\mathbf{W}_j = \mathbf{W}_j^{(d)}\mathbf{P}_j$, with row-orthonormal carriers. Then*

$$|\langle\Delta\mathbf{W}_i, \Delta\mathbf{W}_j\rangle_F| \le \left\|\mathbf{W}_i^{(d)\top}\mathbf{W}_j^{(d)}\right\|_F \cdot \left\|\mathbf{P}_j\mathbf{P}_i^\top\right\|_F.$$

Consequently, $\mathbf{P}_i\mathbf{P}_j^\top = \mathbf{0}$ is sufficient for $\langle\Delta\mathbf{W}_i, \Delta\mathbf{W}_j\rangle_F = 0$. The proof is given in Appendix A.

Catastrophic Forgetting from a Task-Vector View. Prior work on task arithmetic already studies how superposing task updates produces forgetting: when the induced directions are disentangled, adding the updates is approximately an independent superposition, whereas entangled directions let one task's update change another's behavior (Ortiz-Jimenez et al., 2023). Naive merging is exactly this superposition without considering task entanglement, $\Delta\mathbf{W}_{\text{merge}} = \sum_i \Delta\mathbf{W}_i$ (Ilharco et al., 2023). Theorem 1 shows that the cross terms vanish whenever $\mathbf{P}_i\mathbf{P}_j^\top = \mathbf{0}$, and otherwise grow with both subspace overlap and coefficient coupling. Mutual orthogonality of the task vectors is therefore sufficient to remove this entanglement from the overall weight update.

3.2 Adaptive SVD Linear Merging

Linear Merging. Prior work has highlighted the linearity of model fusion (Tang et al., 2024). If coupling is the problem, a merge should isolate a shared term from task residuals rather than add every update in place. A direct sum of LoRA updates mixes transferable directions ($\mathbf{W}_g$) with per-task ones ($\mathbf{W}_{\text{sp}}$). Thus, we merge the QR-standardized archive with a shared-residual rule

$$\Delta\mathbf{W} = \mathbf{W}_g + \mathbf{W}_{\text{sp}} = \Big(\tfrac{1}{T}\sum_{i=1}^{T}\mathbf{W}_i^{(d)}\Big)\mathbf{P}_0 + \sum_{i=1}^{T}\omega_i\mathbf{W}_i^{(d)}(\mathbf{P}_i - \mathbf{P}_0). \tag{1}$$

The merged weights consolidates coefficients on a common carrier $\mathbf{P}_0$; the residual term expresses deviations, without orthogonality assumption. Given archived carriers, the map is linear in $\{\mathbf{W}_i^{(d)}\}$. Each ω_i, and the split of $\mathbf{W}_g$ from $\mathbf{W}_{\text{sp}}$, is set in Appendix D.3.

SVD specifies the carrier. We concatenate all task vectors, apply singular value decomposition (SVD). The rank-r SVD of the stacked carriers is the best coverage of archived directions.

$$\mathbf{X}_T = [\mathbf{P}_1^\top, \ldots, \mathbf{P}_T^\top] = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \qquad \widetilde{\mathbf{P}}_0 = \mathbf{U}_r^\top,$$

Procrustes adaptation. Before the incoming task $\mathcal{T}_{T+1}$, a proper rotation $\mathbf{R}_* \in \mathrm{SO}(r)$ aligns the current $\mathbf{P}_0$ to the last-task-ended carrier, so $\mathbf{P}_0$ is expressed in the coordinates of $\mathbf{P}_T$ rather than in an arbitrary SVD basis.

$$\mathbf{P}_0 = \mathbf{R}_*^\top \widetilde{\mathbf{P}}_0, \qquad \mathbf{R}_* \in \arg\min_{\mathbf{R}\in\mathrm{SO}(r)} \|\mathbf{R}^\top \widetilde{\mathbf{P}}_0 - \mathbf{P}_T\|_F^2, \tag{2}$$

A rank-r SVD specifies a covering subspace, not a unique coordinate frame — $\widetilde{\mathbf{P}}_0$ can be replaced by any $\mathrm{SO}(r)$ rotation of itself. The only sequentially meaningful index is the task just completed: that endpoint $\mathbf{P}_T$ is the carrier we have just obtained and are able to keep its high performance. Then, we seek earlier carriers orthogonal to it to preserve former tasks' performance. In Section 4, we build *task geometry* for that. Thm. 2 shows that this adaptation mitigates forgetting under the geometry.

Theorem 2. *Let $\mathbf{P}$ be a common carrier after Procrustes adaptation and $\mathbf{Q}$ a just-learned task vector. Their difference is $\mathbf{E} = \mathbf{Q} - \mathbf{P}$, which is also the task-specific knowledge. Denote the loss by*

$$\mathcal{L}_g^i = \frac{\|\mathbf{P}\mathbf{E}^\top\|_F^2}{\|\mathbf{E}\|_F^2}, \qquad \mathbf{E} \neq \mathbf{0}. \tag{3}$$

Procrustes alignment minimizes $\|\mathbf{E}\|_F \leq \eta$, which implies $\mathcal{L}_g^i \leq \eta^2/4$, supporting an approximate geometric separation between task-specific knowledge and common carrier. Proof in Appendix C.

4 Chain-updated Learned Task Geometry

The merge above assumes a finished archive. Continual learning must produce that archive online: each task's data is available only during its own training stage, with *no replay* of earlier tasks. Three choices follow: *first*, the next carrier is initialized from the previous one, so training starts at the inherited carrier rather than from a random basis; *second*, a one-sided proxy regularizes displacement of the current task vector from the last-task carrier, which caches a *detached* copy of the current merge and adds it (stop-grad) on the next task's unmerged forward; *third*, after each task the pair $(\mathbf{P}_t, \mathbf{W}_t^{(d)})$ is archived for the eventual merge.

Algorithm 1 ChainLoRA

Require: Base model weights $\mathbf{W}_0$; tasks $\{\mathcal{T}_t\}_{t=1}^T$;
1: Initialize LoRA: kaiming_uniform($\mathbf{A}_1$), $\mathbf{B}_1 = 0$; bake cache empty.
2: **Function** GetProjAndCoeff($\mathbf{A}, \mathbf{B}$):
3: $\quad \mathbf{A}^\top = \mathbf{Q}\mathbf{R}$, $\mathbf{S} \leftarrow \mathrm{diag}\big(\mathrm{sign}(\mathrm{diag}(\mathbf{R}))\big)$, $\widehat{\mathbf{Q}} \leftarrow \mathbf{Q}\mathbf{S}$, $\widehat{\mathbf{R}} \leftarrow \mathbf{S}\mathbf{R}$
4: $\quad \mathbf{P} \leftarrow \widehat{\mathbf{Q}}^\top$, $\mathbf{W}^{(d)} \leftarrow \mathbf{B}\widehat{\mathbf{R}}^\top$
5: $\quad$ **return** $(\mathbf{P}, \mathbf{W}^{(d)})$
6: **for** $t = 1$ to T **do**
7: $\quad$ Train on task $\mathcal{T}_t$, obtain: $\mathbf{A}_t$, $\mathbf{B}_t$.
8: $\quad$ $(\mathbf{P}_t, \mathbf{W}_t{}^{(d)}) \leftarrow$ GetProjAndCoeff($\mathbf{A}_t, \mathbf{B}_t$)
9: $\quad$ Merge the current archive for evaluation; if bake is enabled, cache its detached update for next-task training.
10: $\quad$ Initialize $\mathbf{A}_{t+1} \leftarrow \mathbf{P}_t$, $\mathbf{B}_{t+1} \leftarrow 0$
11: **end for**
12: For $T > 1$, estimate $\mathbf{P}_0$ by SVD and Eq. (2).
13: Compute ω_i, $\mathbf{W}_g$ and $\mathbf{W}_{\mathrm{sp}}$ (Appendix D.3).
14: $\Delta\mathbf{W} \leftarrow (\alpha/r)\mathbf{W}_g + \mathbf{W}_{\mathrm{sp}}$; for $T = 1$, use $\mathbf{B}_1\mathbf{A}_1$.
15: **return** Final model weights $\mathbf{W}_0 + \Delta\mathbf{W}$

Chain-updated initialization. Task 1 starts from a random $\mathbf{P}_1$ and $\mathbf{W}_1^{(d)} = \mathbf{0}$. For $i \geq 2$ we set $\mathbf{P}_i \leftarrow \mathbf{P}_{i-1}$ and $\mathbf{W}_i^{(d)} \leftarrow \mathbf{0}$, then optimize raw factors $\{\mathbf{P}_i, \mathbf{W}_i\}$ on task i and archived them. The next carrier therefore begins at the previous one.

Theorem 3. *Let $\mathbf{E}_i = \mathbf{P}_i - \mathbf{P}_{i-1} \neq \mathbf{0}$, let each earlier increment $\mathbf{E}_j \neq \mathbf{0}$, and write $\widehat{\mathbf{E}}_k = \mathbf{E}_k/\|\mathbf{E}_k\|_F$. Assume $\mathbf{H}_{i-1}$ is row-orthonormal. For every $j < i$, define the new trunk-growth overlap and the earlier-growth trunk gap*

$$\epsilon_i = \big\|\mathbf{H}_{i-1}\widehat{\mathbf{E}}_i^\top\big\|_F, \qquad \delta_{j|i-1} = \big\|\widehat{\mathbf{E}}_j(\mathbf{I} - \mathbf{\Pi}_{i-1})\big\|_F,$$

and the corresponding normalized trunk loss

$$\mathcal{L}_g^i = \frac{\|\mathbf{H}_{i-1}\mathbf{E}_i^\top\|_F}{\|\mathbf{E}_i\|_F} = \epsilon_i. \tag{4}$$

Then

$$\left\|\widehat{\mathbf{E}}_j\widehat{\mathbf{E}}_i^\top\right\|_F \le \mathcal{L}_g^i + \delta_{j|i-1}. \tag{5}$$

Chain-updated one-sided proxy. Before task i, the post-stream reference $\mathbf{P}_0$ is unavailable. The inherited carrier $\mathbf{H}_{i-1} = \mathbf{P}_{i-1}$ is therefore the only trunk the trainer can use. Therefore, we use $\mathbf{H}_{i-1}$ as an effective surrogate.

$$\mathcal{L}_g^i = \sum_{i=1}^{T} \left\|\mathbf{P}_0\Delta\mathbf{P}_i^\top\right\|_F^2 \approx \sum_{i=2}^{T} \left\|\mathbf{H}_{i-1}(\mathbf{P}_i - \mathbf{P}_{i-1})^\top\right\|_F^2 \tag{6}$$

To reason about interference along this chain, write $\mathbf{E}_i = \mathbf{P}_i - \mathbf{P}_{i-1}$ for the local change of the carrier after task i, which is a main carrier rotation. Before task i, $\mathbf{H}_{i-1}$ passes a fixed-rank historical information. By Theorem 3, we could imply that: the orthogonality term of $\Delta\mathbf{P}_i\Delta\mathbf{P}_j^\top$, $\forall i \neq j$ could also be minimized by $\mathcal{L}_g^i$. The proof is given in Appendix B.

5 When LoRA Meets Task Geometry

We now instantiate the geometry with LoRA factors: how $\mathbf{P}_i$ and $\mathbf{W}_i^{(d)}$ are computed, what is optimized online, and how the archive is merged. The overall procedure is Algorithm 1.

5.1 LoRA Corresponds of $\mathbf{P}_i$ and $\mathbf{W}_i^{(d)}$

For task i, we use LoRA factors $\mathbf{A}_i \in \mathbb{R}^{r\times m}$, $\mathbf{B}_i \in \mathbb{R}^{n\times r}$, and define $\Delta\mathbf{W}_i = \mathbf{B}_i\mathbf{A}_i$. To align this with $\Delta\mathbf{W}_i = \mathbf{W}_i^{(d)}\mathbf{P}_i$, we perform thin QR on $\mathbf{A}_i^\top$:

$$\mathbf{A}_i^\top = \mathbf{Q}_i\mathbf{R}_i, \quad \mathbf{S}_i = \mathrm{diag}\big(\mathrm{sign}(\mathrm{diag}(\mathbf{R}_i))\big), \quad \widehat{\mathbf{Q}}_i = \mathbf{Q}_i\mathbf{S}_i, \quad \widehat{\mathbf{R}}_i = \mathbf{S}_i\mathbf{R}_i.$$

Then we have $\mathbf{P}_i = \widehat{\mathbf{Q}}_i^\top$, $\mathbf{W}_i^{(d)} = \mathbf{B}_i\widehat{\mathbf{R}}_i^\top$. So that $\Delta\mathbf{W}_i = \mathbf{W}_i^{(d)}\mathbf{P}_i$ holds exactly.

Improving Parameter Efficiency by Effective-Rank Reduce. We also constrain the Lora parameters with an explicit low-rank surrogate. Since each column of $\mathbf{W}^{(d)}$ and each row of $\mathbf{P}$ corresponds to one latent adaptation direction, shrinking column norms encourages some directions to vanish, thereby reducing the effective rank and minimizing perturbation to the pretrained backbone. Since $\mathbf{P}$ is an orthogonal matrix, its contribution becomes a constant term and does not affect optimization; therefore, the final loss only contains terms w.

$$\Delta\mathbf{W}_i = \mathbf{P}_i\mathbf{W}_i^{(d)} = \sum_{k=1}^{r} p_k w_k^\top, \quad \mathcal{L}_{\text{rank}}^i = \sum_{k=1}^{r}\left(\|p_k\|_2^2 + \|w_k\|_2^2\right) = \sum_{k=1}^{r}\|w_k\|_2^2$$

Initialization. $\mathbf{A}_1 \sim$ Kaiming and $\mathbf{B}_1 = \mathbf{0}$. For $i \ge 2$, we use $\mathbf{A}_i \leftarrow \mathbf{P}_{i-1}$, $\mathbf{B}_i \leftarrow \mathbf{0}$.

Per-task total loss. At step of training task i, we optimize $\mathcal{L}^i = \mathcal{L}_t^i + \lambda_1\mathcal{L}_g^i + \lambda_2\mathcal{L}_{\text{rank}}^i$. After all tasks are learned, updates are merged using Eq. 1.

Merging Phase. According to Eq. 1, we merge the previous tasks parameters and gets the final $\Delta\mathbf{W}$. A key point for merging take effect is the chain-update task geometry. That orthogonality of tasks induced by $\mathcal{L}_g^i$ lets the merge take effect, and chain-updated initialization carries forward the common carrier. The merge depends on both; without them it need not work.

Remarks. Moreover, our formulation can be viewed as a strict generalization of OLoRA-style merging. Specifically, Eq. 1 explicitly decomposes updates into a shared component (through $\mathbf{P}_0$) and a task-specific residual component. When $\mathbf{P}_0 = \mathbf{0}$, the shared term vanishes and Eq. 1 reduces to $\Delta\mathbf{W} = \sum_{i=1}^{T}\mathbf{P}_i\mathbf{W}_i^{(d)} = \sum_{i=1}^{T}\mathbf{A}_i\mathbf{B}_i$, which recovers the OLoRA-like merge pattern that only aggregates task-specific projected updates.

6 EXPERIMENTS

6.1 EXPERIMENTAL SETUP

Base Models. We evaluate our approach across three models: Llama-2-7B-chat, Llama-2-13B-chat, and Llama-3.2-3B. The 3B checkpoint differs from Llama-2 in family as well as size. Although this comparison does not control variables in the strictest sense, it still shows that the method transfers across model families and across sizes. All experiments were conducted on four RTX 4090 GPUs. The learning rate is 2×10^{-4}. Each task in the stream is trained for two epochs by LoRA with bake training as in Appendix D.2. The geometry loss is extended in multi-layers as in Appendix D.1.
Datasets. We consider three continual learning benchmarks: (1) *Standard CL benchmark*: AG News, Amazon Reviews, DBpedia, and Yahoo Answers. (2) *Large number of tasks*: five Standard CL benchmark tasks, four GLUE tasks (MNLI, QQP, RTE, SST-2), five SuperGLUE tasks (WiC, CB, COPA, MultiRC, BoolQ), and IMDB movie reviews. Following O-LoRA (Wang et al., 2023), each task uses 1000 randomly sampled training samples and 500 validation samples per class. (3) *SuperNI Benchmark* (Wang et al., 2022a): SuperNI is an instruction-following mix of generation and classification, unlike the classification streams that follow. A diverse collection of NLP tasks with expert-written instructions, covering dialogue generation, information extraction, question answering, summarization, and sentiment analysis. We follow task selection and ordering in SAPT (Zhao et al., 2024), using 1,000 training instances and 100 for validation/testing per task.
Baselines. We compare our method with the following baselines: (1) *Continual learning baselines:* SeqLoRA: sequentially fine-tunes a single LoRA on multiple tasks without constraints; IncLoRA: incrementally adds a new LoRA per task while freezing previous LoRAs; O-LoRA (Wang et al., 2023); InfLoRA (Liang & Li, 2024); SAPT-LoRA (Zhao et al., 2024); LoRM (Salami et al., 2025); CorDA (knowledge-preserved adaptation) (Yang et al., 2024); MagMax (Marczak et al., 2024). (2) *LoRA merging baselines:* LoRA-LEGO (Zhao et al., 2025); KnOTS (Stoica et al., 2025). (3) *Continual merging baseline:* OPCM (Tang et al., 2025).

Table 1: Final average task scores (%) for Llama-2-7B-chat (Oi=order). Only the replay-free method is compared. Methods are listed in the order described above, while the two merging strategies of LORM are listed separately.

	Standard CL Benchmark				**Large Number of Tasks**				**SuperNI Benchmark**		
Method	**O1**	**O2**	**O3**	**avg**	**O4**	**O5**	**O6**	**avg**	**O1**	**O2**	**avg**
SeqLoRA	73.3	76.2	78.4	76.0	69.1	66.0	71.1	68.7	18.4	26.8	22.6
IncLoRA	75.3	77.3	78.3	77.0	72.2	71.6	73.8	72.5	22.0	25.6	23.8
O-LoRA	76.1	76.3	79.2	77.2	74.0	72.0	74.6	73.5	23.3	28.4	25.9
InfLoRA	78.4	80.4	79.9	79.6	69.4	67.4	72.5	69.8	16.5	22.1	19.3
LoRM-BA	76.0	76.8	78.3	77.0	71.4	69.0	70.3	70.2	25.6	18.7	22.2
LoRM-AB	77.5	74.7	75.9	76.0	71.0	69.5	70.2	70.2	25.6	23.7	24.7
CorDA	78.4	79.3	80.0	79.2	73.4	72.7	74.0	73.4	20.9	16.0	18.5
MagMax	80.1	**80.6**	**80.3**	**80.3**	72.3	73.5	74.5	73.4	15.3	7.0	11.2
OPCM	61.9	62.0	56.7	60.2	51.9	52.8	46.9	50.5	11.6	12.3	12.0
KnOTS	67.9	65.9	70.8	68.2	61.5	60.1	58.0	59.9	**34.6**	30.1	32.4
LoRA-LEGO	68.3	66.0	70.9	68.4	58.8	58.7	53.2	56.9	32.8	26.7	29.8
ChainLoRA (ours)	**80.5**	79.6	80.2	80.1	**77.3**	**75.8**	**74.6**	**75.5**	29.8	**37.8**	**33.8**
(Replayed) SAPT	82.9	81.8	78.7	81.1	84.7	78.9	82.2	81.9	53.2	48.5	50.9

Evaluation metrics. Every suite reports average accuracy (AA) $\frac{1}{T}\sum_{i=1}^{T} a_{i,T}$ after the last task. Classification sets $a_{i,T}$ by exact match; SuperNI is evaluated by Rouge-L first, and AA is the average of those scores. Backward transfer (BWT) $\frac{1}{T-1}\sum_{i=1}^{T-1}(a_{i,T} - a_{i,i})$, is recorded in Appendix E.

6.2 OVERALL RESULTS

On the short Standard stream, our method approaches the best results alongside MagMax and InfLoRA, and replay-augmented SAPT-LoRA leads only slightly. The replay lead is much clearer on

long streams: visible on Large, and largest on SuperNI, where SAPT sits far above the replay-free rows. Even there, ChainLoRA is closest to SAPT, which further shows that the method is strong.

SST-2 is a task in the SuperNI stream. It is learned, then not retained: after later tasks, the score falls to 0 (Figure 4). On O1 it appears early, at position 2, and is already 0 by task 6. On O2 it appears at position 8 and finishes at 0 on the last task. ChainLoRA still sits above the listed merging baselines (KnOTS, LoRA-LEGO). We also believe this task is the opening for our follow-up work.

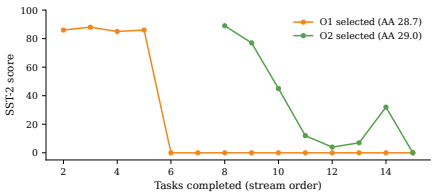

Figure 4: SuperNI SST-2 collapsed.

6.3 Ablation Study

Effect of each module. Table 2 asks which piece matters on which suite. On the short Standard stream only a few tasks are merged, so the merge rule moves the score the most: removing Adaptive SVD drops AA by 5.0. Longer streams already carry some forgetting, and the same removal costs less, 2.3 on Large and 1.3 on SuperNI. The online losses go the other way. They stay small while the tasks are short classification problems, then grow once the stream is longer and the tasks are more mixed: on SuperNI, removing $\mathcal{L}_g^i$ costs 6.2 and removing the rank loss costs 6.3. Keeping chain-updated initialization is necessary on that stream as well. Without it, SuperNI falls by 3.1.

Table 2: Ablation study of each modules as Table 1.

Variant	Standard CL avg	Δ	Large avg	Δ	SuperNI avg	Δ
Full (Table 1 recipe)	80.1	–	75.5	–	33.8	–
w/o $\mathcal{L}_g^i$ (λ_1=0)	79.3	−0.8	74.4	−1.1	27.6	−6.2
w/o $\mathcal{L}_{\text{rank}}^i$ (λ_2=0)	79.0	−1.1	75.4	−0.1	27.5	−6.3
w/o Adaptive SVD Merging	75.1	−5.0	73.2	−2.3	32.5	−1.3
w/o Chain-updated initialization	78.2	−1.9	74.8	−0.7	30.7	−3.1

Varying the merge. Table 3 compares three ways to compose the same archive. Average adds the task updates with equal weight. SVD keeps only the rank-r carrier and leaves that basis where the decomposition put it. Adaptive SVD is the full rule: the same carrier, then rotated onto $\mathbf{P}_T$ by Procrustes. SVD without that alignment collapses. Standard falls to 55.6, SuperNI to 11.3, and Large O4/O5 finish at 0. The SVD factor is a covering subspace, not the coordinate frame of $\mathbf{P}_T$. Without that alignment, carrying the basis straight into the merge makes the result change with the task order, which continual learning cannot accept. Average already keeps a usable score (65.0 / 67.6 / 28.8). Different tasks move in different directions in the update, so a direct sum still carries each of them; the geometric loss further puts those updates in a nearly additive state, which is why the naive sum works. Adaptive SVD remains higher row across the three suites.

Table 3: Merging ablation study. † marks results that ended in catastrophic forgetting (AA = 0).

	Standard CL Benchmark				Large Number of Tasks				SuperNI Benchmark		
Merging	**O1**	**O2**	**O3**	**avg**	**O4**	**O5**	**O6**	**avg**	**O1**	**O2**	**avg**
Average	74.2	65.8	55.0	65.0	67.6	66.7	68.4	67.6	28.4	29.2	28.8
SVD (without Procrustes)	41.6	64.3	60.8	55.6	†	†	54.3	–	10.4	12.2	11.3
Adaptive SVD (ours)	80.5	79.6	80.2	80.1	77.3	75.8	74.6	75.5	29.8	37.8	33.8

Effect of model size. Table 4 lists three base models to check whether the same continual method still works across model family and across size. The comparison does not control every variable in the strictest sense: Llama-3.2-3B is there because it differs from Llama-2 in family as well as size. The method remains effective on all three. Larger size is not uniformly better, since the intrinsic dimension of the model can change as its size grows, and it does not always perform best. At this scale, 7B and 13B still outperform 3B (80.1/75.5/33.8 and 75.5/76.6/23.7 against 72.0/73.3/16.1). A larger parameter space can hold more tasks thus forget less, which is the gain from 3B up to the

Llama-2 models. From 7B to 13B the gain is much smaller, and on the short Standard stream 7B is ahead of 13B (80.1 vs. 75.5), as it is on SuperNI (33.8 vs. 23.7). The order reverses on the Large datasets, and 13B is ahead of 7B (76.6 vs. 75.5), so the two sizes cross. We attribute the difference to depth: 13B has more layers than 7B; loss averaged over layers might lower a given task performance.

Table 4: Results across model size 3B/7B/13B.

	Standard CL Benchmark				Large Number of Tasks				SuperNI Benchmark		
Model	**O1**	**O2**	**O3**	**avg**	**O4**	**O5**	**O6**	**avg**	**O1**	**O2**	**avg**
Llama-3.2-3B	73.6	70.4	72.1	72.0	73.5	72.9	73.4	73.3	14.5	17.8	16.1
Llama-2-7B-chat	80.5	79.6	80.2	80.1	77.3	75.8	74.6	75.5	29.8	37.8	33.8
Llama-2-13B-chat	75.3	74.5	76.7	75.5	78.8	73.5	77.6	76.6	26.4	21.1	23.7

Varying $\mathbf{A}_i$ initialization. Figure 5 isolates the chain-updated carrier with random initialization. The latter is already weaker on the short Standard stream; the gap widens on Large, where each new task otherwise starts from an unrelated basis while the archive is still growing. Because the last carrier is the only shared object the next task can see, it gives a better initialization, and both orthogonalization and learning similar tasks.

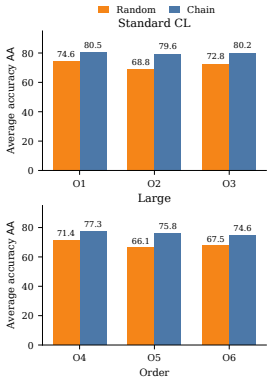

Figure 5: Random vs. Chain init on Standard CL and Large.

6.4 Continual Learning Trajectory

Order averages AA hide where forgetting actually happens. Table 5 is one Standard order, DBpedia→Amazon→Yahoo→AGNews, so the four rows are the four tasks. The three scores are in different times: $a_{i,0}$ is exact match before task i is seen, $a_{i,i}$ is the score just after that task is trained, and $a_{i,T}$ is the score after the whole stream. On the 3B checkpoint, $a_{i,0}$ stays near zero because the model cannot yet emit the label format without training; only after training is the accuracy high enough for a decent prediction on that task, which is why the next column jumps. What the table shows is that forgetting depends on the task: AGNews barely moves once it is learned, while Yahoo can lose a large amount after learning new tasks, especially at 13B. Figure 4 indicates the same point on SuperNI: SST-2 finishes at 0 extremely.

Table 5: Standard CL O1 per-task scores (%) at three sizes (DBpedia→Amazon→Yahoo→AGNews); $a_{i,0}$ unseen, $a_{i,i}$ just after task i, $a_{i,T}$ after the stream, $\Delta{=}a_{i,T}{-}a_{i,i}$.

	3B				7B				13B			
Task	$a_{i,0}$	$a_{i,i}$	$a_{i,T}$	Δ	$a_{i,0}$	$a_{i,i}$	$a_{i,T}$	Δ	$a_{i,0}$	$a_{i,i}$	$a_{i,T}$	Δ
DBpedia	1.4	86.6	84.6	−2.0	60.7	98.8	95.6	−3.2	71.4	98.7	97.2	−1.6
Amazon	3.5	54.8	52.8	−2.0	34.4	62.5	60.0	−2.4	44.6	54.0	59.9	+5.9
Yahoo	0.0	70.4	67.4	−3.0	14.9	74.3	71.2	−3.1	34.8	74.9	54.0	−20.9
AGNews	17.0	89.3	89.4	+0.1	61.9	90.1	90.0	−0.1	56.6	90.0	90.1	+0.0
AA / mean Δ on old		73.6		−2.3		79.2		−2.9		75.3		−5.5

7 Conclusion

We presented **ChainLoRA**, a replay-free continual merging framework built on chain-updated task-vector geometry. The resulting linear merge consolidates transferable directions while preserving task-private knowledge. During the stream, chain-updated initialization copies the previous carrier forward, and a one-sided orthogonality surrogate limits drift from that carrier effectively. We further imposed effective-rank regularization to control long-horizon parameter growth for parameter efficiency. The cost stays constant as the stream lengthens. After training, the archive is absorbed into a single adapter for inference. Experiments show that ChainLoRA is the strongest replay-free method we compare on the long stream and approaches comparable performance on the short stream.

AI USE STATEMENT

In this work, we used generative AI tools to edit the manuscript for readability and LaTeX formatting. We have not used generative AI tools to generate datasets, experimental numbers, or figures. We reviewed the AI-assisted text and checked the mathematical claims against the appendix. We take responsibility for the final content of this work.

ETHICS STATEMENT

This work evaluates publicly available language models on public continual-learning benchmarks. It does not involve human subjects, does not release a new dataset, and does not store user data; the method is replay-free. We are not aware of additional ethics concerns beyond those of ordinary NLP fine-tuning.

REPRODUCIBILITY STATEMENT

All settings are described in the experimental setup or in the design details of Appendix D. The procedure is given as Alg. 1. Anonymous source code will be provided as supplementary material.

REFERENCES

Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3366–3375, 2017.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

Cyprien de Masson D'Autume, Sebastian Ruder, Lingpeng Kong, and Dani Yogatama. Episodic memory in lifelong language learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International conference on artificial intelligence and statistics*, pp. 3762–3773. PMLR, 2020.

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.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023.

Nikitas Karanikolas, Eirini Manga, Nikoletta Samaridi, Eleni Tousidou, and Michael Vassilakopoulos. Large language models versus natural language understanding and generation. In *Proceedings of the 27th Pan-Hellenic Conference on Progress in Computing and Informatics*, pp. 278–290, 2023.

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

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018.

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.

Chengyuan Liu, Yangyang Kang, Shihang Wang, Lizhi Qing, Fubang Zhao, Chao Wu, Changlong Sun, Kun Kuang, and Fei Wu. More than catastrophic forgetting: Integrating general capabilities for domain-specific llms. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7531–7548, 2024.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. *Advances in Neural Information Processing Systems*, 30, 2017.

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.

Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzciński, and Sebastian Cygert. Magmax: Leveraging model merging for seamless continual learning. In *European Conference on Computer Vision*, pp. 379–395. Springer, 2024.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. *Advances in Neural Information Processing Systems*, 36:66727–66754, 2023.

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.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in neural information processing systems*, 32, 2019.

Riccardo Salami, Pietro Buzzega, Matteo Mosconi, Jacopo Bonato, Luigi Sabetta, and Simone Calderara. Closed-form merging of parameter-efficient modules for federated continual learning. In *The Thirteenth International Conference on Learning Representations*, 2025.

Haizhou Shi, Zihao Xu, Hengyi Wang, Weiyi Qin, Wenyuan Wang, Yibin Wang, Zifeng Wang, Sayna Ebrahimi, and Hao Wang. Continual learning of large language models: A comprehensive survey. *ACM Computing Surveys*, 58(5):1–42, 2025.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with SVD to tie the knots. In *The Thirteenth International Conference on Learning Representations*, 2025.

Anke Tang, Li Shen, Yong Luo, Yibing Zhan, Han Hu, Bo Du, Yixin Chen, and Dacheng Tao. Parameter-efficient multi-task model fusion with partial linearization. In *The Twelfth International Conference on Learning Representations*, 2024.

Anke Tang, Enneng Yang, Li Shen, Yong Luo, Han Hu, Bo Du, and Dacheng Tao. Merging models on the fly without retraining: A sequential approach to scalable continual model merging. *arXiv preprint arXiv:2501.09522*, 2025.

Kai Tong, Kang Pan, Xiao Zhang, Erli Meng, Run He, Yawen Cui, Nuoyan Guo, and Huiping Zhuang. Any-ssr: How recursive least squares works in continual learning of large language model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3047–3057, 2025.

Vinay Kumar Verma, Kevin J Liang, Nikhil Mehta, Piyush Rai, and Lawrence Carin. Efficient feature transformations for discriminative and generative continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13865–13875, 2021.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10658–10671, 2023.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. *arXiv preprint arXiv:2204.07705*, 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.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pp. 23965–23998. PMLR, 2022.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen Leon Song, Jianlong Wu, Liqiang Nie, and Bernard Ghanem. CorDA: Context-oriented decomposition adaptation of large language models for task-aware parameter-efficient fine-tuning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Jinghan Zhang, Junteng Liu, Junxian He, et al. Composing parameter-efficient modules with arithmetic operation. *Advances in Neural Information Processing Systems*, 36:12589–12610, 2023.

Weixiang Zhao, Shilong Wang, Yulin Hu, Yanyan Zhao, Bing Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. Sapt: A shared attention framework for parameter-efficient continual learning of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 11641–11661, 2024.

Ziyu Zhao, Tao Shen, Didi Zhu, Zexi Li, Jing Su, Xuwu Wang, and Fei Wu. Merging loRAs like playing LEGO: Pushing the modularity of loRA to extremes through rank-wise clustering. In *The Thirteenth International Conference on Learning Representations*, 2025.

Heming Zou, Yunliang Zang, Wutong Xu, Yao Zhu, and Xiangyang Ji. Flylora: Boosting task decoupling and parameter efficiency via implicit rank-wise mixture-of-experts. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Appendix

This appendix is organized in four blocks. **Theory**: coefficient-weighted task-vector interaction (Appendix A), the normalized interference bound (Appendix B), and fourth-order overlap with the similar-task bound (Appendix C). **Training and merge** (Appendix D): layer-averaged geometric and rank losses (Appendix D.1), inter-task bake (Appendix D.2), and distance-dependent residual weights (Appendix D.3). **Backward transfer and running curves** (Appendix E): final backward transfer (Appendix E.1) and running backward transfer and average accuracy (Appendix E.2).

Contents of the Appendix

Theory.

A Proof of Theorem 1 . . . 13

B Proof of Theorem 3 . . . 14

C Proof of Theorem 2 . . . 14

Setup details.

D Implementation defaults . . . 16

D.1 Training-objective correspondence . . . 16

D.2 Bake Training . . . 16

D.3 Residual weights . . . 16

Additional experiments.

E Additional experiments overview . . . 17

E.1 Backward transfer . . . 17

E.2 Running average accuracy . . . 17

A Proof of Theorem 1

Lemma 1 (Frobenius Cauchy–Schwarz)**.** *For real matrices* $\mathbf{A}, \mathbf{B}$ *of the same shape,*

$$\left|\langle \mathbf{A}, \mathbf{B}\rangle_F\right| \le \|\mathbf{A}\|_F \|\mathbf{B}\|_F.$$

If $\mathbf{A}$ *and* $\mathbf{B}$ *can be multiplied, then* $\left|\mathrm{tr}(\mathbf{AB})\right| \le \|\mathbf{A}\|_F \|\mathbf{B}\|_F$.

Proof. Write the entries as vectors. The first bound is the Euclidean Cauchy–Schwarz inequality

$$\left|\sum_{ab} \mathbf{A}_{ab}\mathbf{B}_{ab}\right| \le \Big(\sum_{ab} \mathbf{A}_{ab}^2\Big)^{1/2} \Big(\sum_{ab} \mathbf{B}_{ab}^2\Big)^{1/2}.$$

For the second, $\mathrm{tr}(\mathbf{AB}) = \langle \mathbf{A}^\top, \mathbf{B}\rangle_F$ and $\|\mathbf{A}^\top\|_F = \|\mathbf{A}\|_F$. □

Proof of Theorem 1. By cyclicity of the trace,

$$\begin{aligned}\langle \Delta\mathbf{W}_i, \Delta\mathbf{W}_j\rangle_F &= \mathrm{tr}\big(\mathbf{P}_i^\top \mathbf{W}_i^{(d)\top} \mathbf{W}_j^{(d)} \mathbf{P}_j\big) \\ &= \mathrm{tr}\big(\mathbf{W}_i^{(d)\top} \mathbf{W}_j^{(d)} \mathbf{P}_j \mathbf{P}_i^\top\big).\end{aligned}$$

Lemma 1 then gives

$$\left|\langle \Delta\mathbf{W}_i, \Delta\mathbf{W}_j\rangle_F\right| \le \left\|\mathbf{W}_i^{(d)\top} \mathbf{W}_j^{(d)}\right\|_F \cdot \left\|\mathbf{P}_j \mathbf{P}_i^\top\right\|_F.$$

If $\mathbf{P}_i \mathbf{P}_j^\top = \mathbf{0}$, then $\left|\langle \Delta\mathbf{W}_i, \Delta\mathbf{W}_j\rangle_F\right|$ is also zero, and the update inner product vanishes. □

B Proof of Theorem 3

Throughout, $\mathbf{E}_i = \mathbf{P}_i - \mathbf{P}_{i-1}$ is the local carrier increment after task i, $\widehat{\mathbf{E}}_i = \mathbf{E}_i / \|\mathbf{E}_i\|_F$, and the current historical carrier is $\mathbf{H}_{i-1} = \mathbf{P}_{i-1}$ with projector $\mathbf{\Pi}_{i-1} = \mathbf{H}_{i-1}^\top \mathbf{H}_{i-1}$. The carrier is the rank-r carrier passed along the chain.

Proof of Theorem 3. For convenience, denote $\mathcal{L}_g^i$ as ϵ_i:

$$\epsilon_i = \left\|\mathbf{H}_{i-1}\widehat{\mathbf{E}}_i^\top\right\|_F = \frac{\|\mathbf{H}_{i-1}\mathbf{E}_i^\top\|_F}{\|\mathbf{E}_i\|_F}.$$

Thus the argument never assumes $\|\mathbf{E}_i\|_F = 1$; only $||\widehat{\mathbf{E}}_i||_F = 1$ is unit-Frobenius.

Insert $\mathbf{I} = \mathbf{\Pi}_{i-1} + (\mathbf{I} - \mathbf{\Pi}_{i-1})$:

$$\widehat{\mathbf{E}}_j\widehat{\mathbf{E}}_i^\top = \widehat{\mathbf{E}}_j\mathbf{\Pi}_{i-1}\widehat{\mathbf{E}}_i^\top + \widehat{\mathbf{E}}_j(\mathbf{I} - \mathbf{\Pi}_{i-1})\widehat{\mathbf{E}}_i^\top. \tag{7}$$

For the first term, substitute $\mathbf{\Pi}_{i-1} = \mathbf{H}_{i-1}^\top\mathbf{H}_{i-1}$:

$$\widehat{\mathbf{E}}_j\mathbf{\Pi}_{i-1}\widehat{\mathbf{E}}_i^\top = \left(\widehat{\mathbf{E}}_j\mathbf{H}_{i-1}^\top\right)\left(\mathbf{H}_{i-1}\widehat{\mathbf{E}}_i^\top\right).$$

Lemma 1 then gives:

$$\left\|\widehat{\mathbf{E}}_j\mathbf{\Pi}_{i-1}\widehat{\mathbf{E}}_i^\top\right\|_F \le \left\|\widehat{\mathbf{E}}_j\mathbf{H}_{i-1}^\top\right\|_F\left\|\mathbf{H}_{i-1}\widehat{\mathbf{E}}_i^\top\right\|_F.$$

The second factor is ϵ_i by definition. The first factor is at most one:

$$\left\|\widehat{\mathbf{E}}_j\mathbf{H}_{i-1}^\top\right\|_F \le \left\|\widehat{\mathbf{E}}_j\right\|_F\left\|\mathbf{H}_{i-1}\right\|_F = 1,$$

The first term is therefore at most ϵ_i; For the second term the same product bound gives

$$\left\|\widehat{\mathbf{E}}_j(\mathbf{I} - \mathbf{\Pi}_{i-1})\widehat{\mathbf{E}}_i^\top\right\|_F \le \left\|\widehat{\mathbf{E}}_j(\mathbf{I} - \mathbf{\Pi}_{i-1})\right\|_F\left\|\widehat{\mathbf{E}}_i^\top\right\|_F = \left\|\widehat{\mathbf{E}}_j(\mathbf{I} - \mathbf{\Pi}_{i-1})\right\|_F.$$

Since $\|\widehat{\mathbf{E}}_i^\top\|_F = 1$, so the second term is at most $\delta_{j|i-1}$. The triangle inequality yields $\|\widehat{\mathbf{E}}_j\widehat{\mathbf{E}}_i^\top\|_F \le \epsilon_i + \delta_{j|i-1} = \mathcal{L}_g^i + \delta_{j|i-1}$. □

Remarks The part of the bound cannot remove is $\delta_{j|i-1}$. It is the piece of $\widehat{\mathbf{E}}_j$ outside the common carrier, and it is the obstruction to orthogonality when that carrier is the rank-r carrier of every task. At task i this carrier is not yet available. The inherited carrier $\mathbf{H}_{i-1} = \mathbf{P}_{i-1}$ is the best approximation on hand, so $\delta_{j|i-1}$ is treated as the incompressible part between the two tasks under that approximation. When that loss $\mathcal{L}_g^i$ is close to zero, the interference bound reduces to $\delta_{j|i-1}$: the two normalized increments have been driven as close to orthogonal as this approximation of the carrier allows.

C Proof of Theorem 2

A rotation of the basis inside a fixed subspace changes the numerical entries of a carrier even though the subspace itself has not moved. If that basis is not brought to the angle at which orthogonality is measured, carriers can already be orthogonal but look coupled, because only the gauge differs.

Forcing the penalty in that mismatched gauge then rotates an already acceptable pair and can itself induce forgetting. The alignment below removes this rotational factor by rotating the later carrier onto the earlier one, so the diagnostic sees only what is left after that gauge is fixed.

Lemma 2 (LoRA overlap)**.** *In the LoRA factors of Section 5.1, $\mathbf{A} \in \mathbb{R}^{r\times m}$ and $\mathbf{B} \in \mathbb{R}^{n\times r}$. Thin QR of $\mathbf{A}^\top$ returns a carrier $\mathbf{P} \in \mathbb{R}^{r\times m}$ with $\mathbf{P}\mathbf{P}^\top = \mathbf{I}_r$, so $\mathbf{P}$ lies on the Stiefel manifold*

$$\mathrm{St}(r, m) = \{\mathbf{P} \in \mathbb{R}^{r\times m} : \mathbf{P}\mathbf{P}^\top = \mathbf{I}_r\}.$$

Left multiplication by an orthogonal $r \times r$ matrix stays on $\mathrm{St}(r, m)$ and does not change the row space: it only rechooses the basis inside the same subspace. The coefficient is $\mathbf{W}^{(d)} = \mathbf{B}\widehat{\mathbf{R}}^\top$. For a second carrier $\mathbf{Q}$ of the same shape and $\mathbf{E} = \mathbf{Q} - \mathbf{P}$,

$$\mathbf{P}\mathbf{E}^\top \in \mathbb{R}^{r\times r}.$$

Lemma 3 (Symmetric splitting). *Let* $\mathbf{M}$ *be a square matrix of the same shape as* $\mathbf{P}\mathbf{E}^\top$ *in Lemma 2, hence* $r \times r$.

$$\mathrm{sym}(\mathbf{M}) = \tfrac{1}{2}(\mathbf{M} + \mathbf{M}^\top), \qquad \mathrm{skew}(\mathbf{M}) = \tfrac{1}{2}(\mathbf{M} - \mathbf{M}^\top).$$

Then $\mathbf{M} = \mathrm{sym}(\mathbf{M}) + \mathrm{skew}(\mathbf{M})$*, and the two parts are orthogonal.*

Denote the raw squared overlap by

$$\mathcal{L} = \|\mathbf{P}(\mathbf{Q} - \mathbf{P})^\top\|_F^2 = \|\mathbf{P}\mathbf{E}^\top\|_F^2, \qquad \mathbf{E} = \mathbf{Q} - \mathbf{P}. \tag{8}$$

The orthogonality loss of Eq. (3) is this square divided by the squared distance:

$$\mathcal{L}_g^i = \frac{\mathcal{L}}{\|\mathbf{E}\|_F^2}, \qquad \mathbf{E} \neq \mathbf{0}. \tag{9}$$

We therefore convert the claim to be verified into the following two statements.

(i) We prove that $\mathcal{L}$ is the fourth-order bound: $\frac{1}{4r}\|\mathbf{E}\|_F^4 \le \mathcal{L} \le \frac{1}{4}\|\mathbf{E}\|_F^4$.

(ii) Eq. (9) then makes $\mathcal{L}_g^i$ the second-order bound: $\frac{1}{4r}\|\mathbf{E}\|_F^2 \le \mathcal{L}_g^i \le \frac{1}{4}\|\mathbf{E}\|_F^2$.

Proof of Theorem 2. Let $\mathbf{P}, \mathbf{Q} \in \mathbb{R}^{r\times m}$ be row-orthonormal and set $\mathbf{E} = \mathbf{Q} - \mathbf{P}$, with $0 < \|\mathbf{E}\|_F \le \eta$. By Lemma 2, $\mathbf{P}\mathbf{E}^\top$ is $r \times r$.

Procrustes alignment seeks largest $\mathrm{tr}(\mathbf{P}\mathbf{Q}^\top\mathbf{R})$**.** The rotation in Eq. (2) is the first restriction: minimize $\|\mathbf{R}^\top\mathbf{Q} - \mathbf{P}\|_F^2$ over $\mathbf{R} \in \mathrm{SO}(r)$. Expand the square in the Frobenius inner product:

$$\begin{aligned}\|\mathbf{R}^\top\mathbf{Q} - \mathbf{P}\|_F^2 &= \|\mathbf{R}^\top\mathbf{Q}\|_F^2 + \|\mathbf{P}\|_F^2 - 2\langle\mathbf{R}^\top\mathbf{Q}, \mathbf{P}\rangle_F \\ &= \|\mathbf{Q}\|_F^2 + \|\mathbf{P}\|_F^2 - 2\,\mathrm{tr}(\mathbf{P}\mathbf{Q}^\top\mathbf{R}).\end{aligned}$$

The second uses that $\mathbf{R}$ is orthogonal, so $\|\mathbf{R}^\top\mathbf{Q}\|_F = \|\mathbf{Q}\|_F$, together with

$$\langle\mathbf{R}^\top\mathbf{Q}, \mathbf{P}\rangle_F = \mathrm{tr}\big(\mathbf{P}(\mathbf{R}^\top\mathbf{Q})^\top\big) = \mathrm{tr}(\mathbf{P}\mathbf{Q}^\top\mathbf{R}).$$

Making the square smaller is therefore the same as making $\mathrm{tr}(\mathbf{P}\mathbf{Q}^\top\mathbf{R})$ larger.

The largest trace: two situations. Write the aligned carrier as $\mathbf{Q}^\star = \mathbf{R}^\top\mathbf{Q}$. Its cross-Gram is $\mathbf{S} = \mathbf{P}(\mathbf{Q}^\star)^\top = \mathbf{P}\mathbf{Q}^\top\mathbf{R}$.

Take the SVD of the unaligned cross-Gram, $\mathbf{P}\mathbf{Q}^\top = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. Then $\mathrm{tr}(\mathbf{S}) = \mathrm{tr}(\mathbf{\Sigma}\mathbf{V}^\top\mathbf{R}\mathbf{U})$, so a rotation maximizes the trace by driving $\mathbf{V}^\top\mathbf{R}\mathbf{U}$ toward $\mathbf{I}$. The factors $\mathbf{V}$ and $\mathbf{U}$ are that alignment; the two cases below are only whether $\mathbf{V}\mathbf{U}^\top$ is already a rotation.

(i) When $\det(\mathbf{V}\mathbf{U}^\top) = 1$, $\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top\mathbf{V}\mathbf{U}^\top = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^\top$, so $\mathbf{S}^\top = \mathbf{S}$.

(ii) When $\det(\mathbf{V}\mathbf{U}^\top) = -1$, the factor $\mathbf{V}\mathbf{U}^\top$ is not a rotation. Set $\mathbf{D} = \mathrm{diag}(1, \ldots, 1, -1)$ and $\mathbf{R} = \mathbf{V}\mathbf{D}\mathbf{U}^\top$. Then $\det(\mathbf{R}) = 1$. With this rotation,

$$\mathbf{S} = \mathbf{P}(\mathbf{Q}^\star)^\top = \mathbf{U}\mathbf{\Sigma}\mathbf{D}\mathbf{U}^\top = \mathbf{U}\,\mathrm{diag}(\sigma_1, \ldots, \sigma_{r-1}, -\sigma_r)\mathbf{U}^\top.$$

The new diagonal matrix is still equal to its transpose, so again $\mathbf{S}^\top = \mathbf{S}$. The displacement used below is the difference from this aligned carrier,

$$\mathbf{E} = \mathbf{Q}^\star - \mathbf{P}, \qquad 0 < \|\mathbf{E}\|_F \le \eta.$$

Expand the product and use $\mathbf{P}\mathbf{P}^\top = \mathbf{I}_r$:

$$\mathbf{P}\mathbf{E}^\top = \mathbf{P}(\mathbf{Q}^\star - \mathbf{P})^\top = \mathbf{P}(\mathbf{Q}^\star)^\top - \mathbf{P}\mathbf{P}^\top = \mathbf{S} - \mathbf{I}_r.$$

Both $\mathbf{S}$ and $\mathbf{I}_r$ are symmetric, so $\mathbf{P}\mathbf{E}^\top$ is symmetric and $\mathrm{skew}(\mathbf{P}\mathbf{E}^\top) = \frac{1}{2}\big(\mathbf{P}\mathbf{E}^\top - (\mathbf{P}\mathbf{E}^\top)^\top\big) = \mathbf{0}$.

Induced symmetry part. Row orthonormality of $\mathbf{Q} = \mathbf{P} + \mathbf{E}$ gives $\mathbf{P}\mathbf{E}^\top + \mathbf{E}\mathbf{P}^\top = -\mathbf{E}\mathbf{E}^\top$. The left-hand side is twice the symmetric part, so $\mathrm{sym}(\mathbf{P}\mathbf{E}^\top) = -\frac{1}{2}\mathbf{E}\mathbf{E}^\top$.

Fourth-order bound. The skew part is already zero, so $\mathbf{P}\mathbf{E}^\top$ equals its symmetric part:

$$\mathbf{P}\mathbf{E}^\top = -\tfrac{1}{2}\mathbf{E}\mathbf{E}^\top, \qquad \|\mathbf{P}\mathbf{E}^\top\|_F^2 = \tfrac{1}{4}\|\mathbf{E}\mathbf{E}^\top\|_F^2. \tag{10}$$

If $s_1, \ldots, s_q$ are the nonzero singular values of $\mathbf{E}$, with $q \le r$, then $\|\mathbf{E}\mathbf{E}^\top\|_F^2 = \sum_{k=1}^q s_k^4$. Lemma 1 applied to $(s_1^2, \ldots, s_q^2)$ and $(1, \ldots, 1)$ gives $\left(\sum_{k=1}^q s_k^2\right)^2 \le q \sum_{k=1}^q s_k^4$. Since $q \le r$, we have the *left bound*; the cross terms in $\left(\sum_k s_k^2\right)^2$ are nonnegative, it holds the *right bound*:

$$\frac{1}{r}\Big(\sum_{k=1}^q s_k^2\Big)^2 \le \sum_{k=1}^q s_k^4 \le \Big(\sum_{k=1}^q s_k^2\Big)^2.$$

Substitution bounds the raw overlap by $\mathcal{L} = \frac{1}{4}\|\mathbf{E}\mathbf{E}^\top\|_F^2 = \frac{1}{4}\sum_{k=1}^q s_k^4$:

$$\frac{1}{4r}\|\mathbf{E}\|_F^4 \le \mathcal{L} \le \frac{1}{4}\|\mathbf{E}\|_F^4. \tag{11}$$

Second-order bound. Eq. (9) divides $\mathcal{L}$ by $\|\mathbf{E}\|_F^2$. Dividing Eq. (11) by the same factor gives

$$\frac{1}{4r}\|\mathbf{E}\|_F^2 \le \mathcal{L}_g^i \le \frac{1}{4}\|\mathbf{E}\|_F^2. \tag{12}$$

Read back to the theorem itself. Since $\|\mathbf{E}\|_F \le \eta$, we have $\mathcal{L}_g^i \le \frac{1}{4}\|\mathbf{E}\|_F^2 \le \frac{1}{4}\eta^2$. □

D Design Details in LoRA

D.1 Training-Objective Correspondence

With L adapted layers, training uses the main-text losses and averages them over layers. The geometric term is

$$\mathcal{L}_g^i = \frac{1}{L}\sum_{\ell=1}^{L} \frac{\|\mathbf{H}_{i-1,\ell}\mathbf{E}_{i,\ell}^\top\|_F}{\|\mathbf{E}_{i,\ell}\|_F + \gamma}, \tag{13}$$

where $\mathbf{E}_{i,\ell} = \mathbf{P}_{i,\ell} - \mathbf{P}_{i-1,\ell}$ and $\mathbf{H}_{i-1,\ell} = \mathbf{P}_{i-1,\ell}$. The constant is $\gamma = 10^{-8}$, added so the denominator cannot be zero. The rank term is the same penalty on each LoRA layer,

$$\mathcal{L}_{\text{rank}}^i = \frac{1}{L}\sum_{\ell=1}^{L}\frac{1}{r}\sum_{k=1}^{r}\sqrt{\|\mathbf{B}_{i,\ell,:,k}\|_2^2 + \varepsilon}, \tag{14}$$

where $\varepsilon = 10^{-12}$ keeps the square root finite when a column vanishes. Because $\mathbf{W}^{(d)} = \mathbf{B}\widehat{\mathbf{R}}^\top$, these columns are the per-layer form of the w_k in the main text.

D.2 Bake Training

During training, the adapter is merged into the weights after each task. A separate adapter for every task would take more storage as the stream grows. The next LoRA is then trained above the former merged update, so earlier tasks remain in the forward pass.

Training forward. On task $i+1$, suppressing transpose conventions and dropout,

$$y = \mathbf{W}_0 x + \text{stopgrad}(\Theta_i)x + \mathbf{B}_{i+1}\mathbf{A}_{i+1}x,$$

where Θ_i is the merged update of the tasks seen so far. This matches Algorithm 1.

D.3 Residual Weights

Equation (1) writes the merged update as a shared trunk plus weighted residuals, without explaining ω_i. For $T > 1$, considering the distance from the $\mathbf{P}_0$,

$$\mathbf{D}_i = \mathbf{P}_i - \mathbf{P}_0, \qquad d_i = \|\mathbf{D}_i\|_2, \qquad \bar{d} = \max\Big\{\frac{1}{T}\sum_{i=1}^{T} d_i,\ 10^{-12}\Big\},$$

and dacays in proportion to how far it sits from $\mathbf{P}_0$:

$$\omega_i = \frac{1}{1 + d_i/\bar{d}}, \qquad \eta = 1. \tag{15}$$

The two branches of Eq. (1) are then

$$\mathbf{W}_g = \Big(\frac{1}{T}\sum_{i=1}^{T} \mathbf{W}_i^{(d)}\Big)\mathbf{P}_0, \qquad \mathbf{W}_{\text{sp}} = \sum_{i=1}^{T} \omega_i \mathbf{W}_i^{(d)} \mathbf{D}_i, \tag{16}$$

and the absorbed update scales only the trunk by the LoRA factor,

$$\Delta\mathbf{W} = \frac{\alpha}{r}\mathbf{W}_g + \mathbf{W}_{\text{sp}}.$$

E Additional Experiments

E.1 Backward Transfer

Table 6: Stored final BWT for the model-size grid in Table 4. The lower value indicates better performance in hindering forgetting.

	Standard CL				Large number of tasks				SuperNI		
Model	O1	O2	O3	avg	O4	O5	O6	avg	O1	O2	avg
Llama-3.2-3B	−0.1	−7.0	−5.7	−4.3	−0.3	−2.9	−4.9	−2.7	−16.7	−24.6	−20.6
Llama-2-7B-chat	−0.6	−1.6	−1.9	−1.4	+0.4	−4.4	−5.7	−3.2	−10.8	−7.1	−8.9
Llama-2-13B-chat	−5.5	−7.1	−6.7	−6.4	−10.6	−22.0	−13.7	−15.4	−23.7	−30.8	−27.3

E.2 Running average accuracy

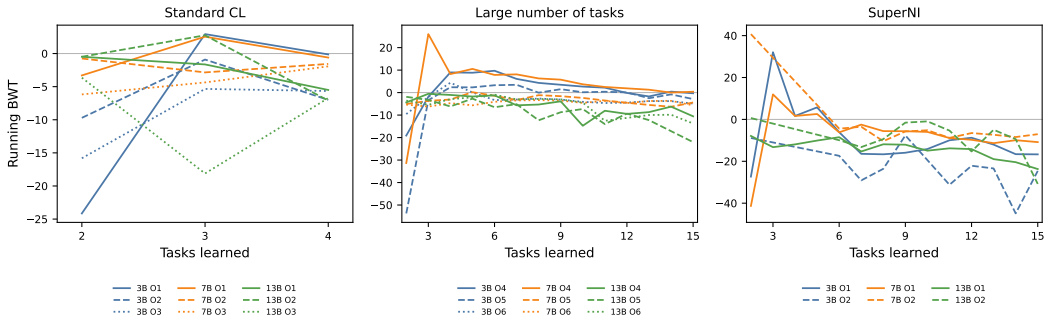

Figure 6: Running BWT after each completed task, for the runs in Table 6.

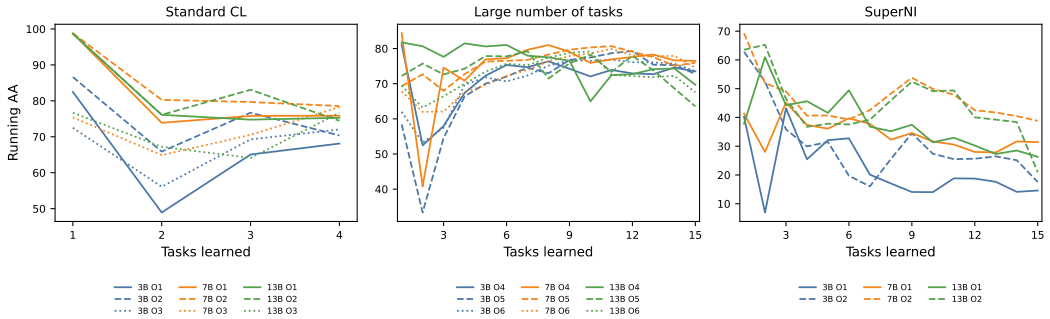

Figure 7: Running AA after each completed task, for the runs in Table 6. At step t, AA_t averages one score for each task seen so far.