Geometry Conflict: Explaining and Controlling Forgetting in LLM Continual Post-Training

Yuanyi Wang[1*‡] **Yifan Yang**[1*] **Su Lu**[1], **Yanggan Gu**[1], **Pengkai Wang**[1], **Wenjun Wang**[1], **Zhaoyi Yan**[2], **Congkai Xie**[2], **Jianmin Wu**[1], **Jialun Cao**[3], **Shing-Chi Cheung**[3], **Hongxia Yang**[1,2,4†‡]
[1]The Hong Kong Polytechnic University, PolyU [2]InfiX.ai, Singapore
[3]The Hong Kong University of Science and Technology, HKUST
[4]Research Institute of Generative Artificial Intelligence

Code: https://github.com/wyy-code/GCWM

Abstract

Continual post-training aims to extend large language models (LLMs) with new knowledge, skills, and behaviors, yet it remains unclear when sequential updates enable capability transfer and when they cause catastrophic forgetting. Existing methods mitigate forgetting through sequential fine-tuning, replay, regularization, or model merging, but offer limited criteria for determining when incorporating new updates is beneficial or harmful. In this work, we study LLM continual post-training through three questions: *What drives forgetting? When do sequentially acquired capabilities transfer or interfere? How can compatibility be used to control update integration?* We address these questions through *task geometry*: we represent each post-training task by its parameter update and study the covariance geometry induced by the update. Our central finding is that: **forgetting can be considered as a state-relative update-integration failure, it arises when the covariance geometries induced by tasks misalign with the geometry of the evolving model state**. Sequential updates transfer when they remain compatible with the model state shaped by previous updates, and interfere when state-relative geometry conflict becomes high. Motivated by this finding, we propose **G**eometry-**C**onflict **W**asserstein **M**erging (GCWM), a data-free update-integration method that constructs a shared Wasserstein metric via Gaussian Wasserstein barycenters and uses geometry conflict to gate geometry-aware correction. Across Qwen3 0.6B–14B on domain-continual and capability-continual settings, GCWM consistently outperforms data-free baselines, improving retention and final performance without replay data. These results identify geometry conflict as both an explanatory signal for forgetting and a practical control signal for LLM continual post-training.

1 Introduction

Continual post-training is becoming an increasingly important paradigm for extending large language models (LLMs) [1, 2]. Rather than learning jointly over all desired capabilities or data, a model is expected to learn through a sequence of post-training stages, each targeting a new domain [3, 4], skill [5, 6], or behavior [7, 8]. This process is natural for real scenarios as capabilities are introduced incrementally. However, sequential post-training faces a fundamental challenge: learning a new

*Equal contribution.
†Corresponding author.
‡yuan-yi.wang@connect.polyu.hk, hongxia.yang@polyu.edu.hk

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

task undermines the knowledge acquired from previous ones, a phenomenon known as catastrophic forgetting [9, 10], often driven by interference between sequential parameter updates.

Existing approaches can be broadly categorized into four classes: sequential fine-tuning [11, 12], replay-based methods that revisit past data [13, 14], regularization methods that constrain update drift [15, 16], or model merging strategies that combine task-specific adaptations [17, 18]. These approaches have led to important progress, but they still lack a principled account of *task compatibility* in continual post-training. As a result, they often struggle to answer a central practical question: when should new parameter updates be strongly integrated into the current model, and when should such integration be restrained? This issue is particularly pronounced for LLMs, where tasks are highly heterogeneous, post-training objectives differ substantially, and the same update magnitude can lead to very different retention outcomes [19].

To address this problem, we study LLM continual post-training through three questions: *What drives forgetting? When do sequentially acquired capabilities transfer or interfere? How can compatibility be used to control update integration?* We answer these questions through a task-geometry view of post-training updates. Specifically, we represent each task by its parameter update and study the induced covariance geometry, which captures not only update magnitude but also the subspaces and spectral structure through which a task changes the model. We define *geometry conflict* as a normalized Bures–Wasserstein discrepancy [20] between task-induced covariance geometries in a shared space, and use its state-relative form to measure compatibility with the evolving LLM state.

Our analysis (Sec. 3) across Qwen3 scales and continual strategies compares geometry conflict with update norm, subspace alignment ratio [21], and gradient conflict [22]. It reveals a central mechanism: forgetting can be considered as a state-relative update-integration failure, it arises when the covariance geometries induced by tasks misalign with the geometry of the evolving model state, whereas transfer occurs when new updates remain compatible with the state shaped by previous updates. This explains why raw update norm and isolated pairwise compatibility are insufficient, and why geometry conflict serves as a natural signal for controlling sequential update integration.

Motivated by this finding, we propose **G**eometry-**C**onflict **W**asserstein **M**erging (GCWM), a data-free update-integration method for LLM continual post-training. GCWM constructs task-induced covariance geometry, builds a shared Wasserstein metric via Gaussian Wasserstein barycenters, and uses geometry conflict to gate geometry-aware correction, which allows GCWM to perform compatibility-controlled update integration. We further provide theoretical support showing that the induced loss change is controlled by geometry conflict and gated merge displacement.

Across *domain-continual* and *capability-continual* settings, GCWM consistently improves retention and final performance over data-free baselines without replay data. On Qwen3 models from 0.6B to 14B, GCWM remains the strongest data-free update-integration method across scales, showing that geometry conflict is useful not only as an explanatory signal for forgetting but also as a practical control signal for continual post-training. In summary, our contributions are summarized as follows:

(i) We develop a task-geometry analysis of LLM continual post-training and show that forgetting is better explained as a state-relative update-integration failure, beyond update norm and isolated pairwise compatibility.
(ii) We introduce *geometry conflict*, a Bures–Wasserstein distance over task-induced covariance geometries, and identify it as both an explanatory signal for forgetting and a compatibility signal for update integration, complementing existing subspace alignment ratio and gradient conflict.
(iii) We propose Geometry-Conflict Wasserstein Merging, a data-free update-integration method that constructs a shared Wasserstein metric and gates geometry-aware correction by layer-wise conflict.
(iv) We derive a conflict-controlled theory linking GCWM's relative loss to geometry conflict and gated merge displacement, and validate GCWM on Qwen3 0.6B–14B across domain- and capability-continual settings, improving final performance over data-free baselines without replay data.

2 Preliminary

2.1 Problem Setup

We study continual post-training for LLMs. Starting from a pretrained model with parameters θ_{pre}, the model is adapted through a sequence of tasks $\mathcal{T} = \{T_1, \ldots, T_K\}$, where each task introduces a

new domain, skill, or behavior. For task T_t, we denote its task-specific update by

$$\Delta_t = \theta_t - \theta_{\text{pre}},$$

where θ_t is the model adapting to T_t. We use these task updates, which may be parameter-efficient or full-model updates, as the basic objects for analyzing in LLM continual post-training.

2.2 Task Geometry and Compatibility Signals

A task update is not fully characterized by its norm: two updates with similar magnitude can affect different subspaces and induce different forgetting behavior. For a layer ℓ, let $\Delta_t^{(\ell)} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$ denote the update matrix of task T_t. Motivated by the task vector [23], we define task geometry as:

$$C_t^{(\ell)} = (\Delta_t^{(\ell)})^\top \Delta_t^{(\ell)},$$

which captures the dominant directions of the update. Let $V_i^{(\ell)}, S_i^{(\ell)}$ be retained SVD factors and $Q^{(\ell)} = \text{orth}([V_i^{(\ell)}, V_j^{(\ell)}])$ their shared union basis. The projected geometry $B_i^{(\ell)} = (Q^{(\ell)})^\top V_i^{(\ell)} (S_i^{(\ell)})^2 (V_i^{(\ell)})^\top Q^{(\ell)} + \lambda I$ is the regularized projection of the retained component of $C_i^{(\ell)}$, with $\lambda > 0$; $B_j^{(\ell)}$ is constructed analogously. Here, d_{B} denotes the Bures–Wasserstein distance between positive-semidefinite matrices, and $\text{tr}(A) = \sum_k A_{kk}$ is the matrix trace. To compare two tasks, we project them into a shared basis and measure their discrepancy using a normalized Bures–Wasserstein distance [20]:

$$\gamma_{ij}^{(\ell)} = \frac{d_{\text{B}}^2(B_i^{(\ell)}, B_j^{(\ell)})}{\text{tr}(B_i^{(\ell)}) + \text{tr}(B_j^{(\ell)}) + \varepsilon},$$

where $B_i^{(\ell)}$ and $B_j^{(\ell)}$ are the projected geometries. We refer to $\gamma_{ij}^{(\ell)}$ as *geometry conflict*. Lower values indicate more compatible task-induced geometries. In Sec. 3, we compare geometry conflict with three standard diagnostics: update norm, subspace alignment ratio (SAR) [24], and gradient cosine conflict [25]. State-relative variants replace one task update with the current continual-training state. Full metric definitions and aggregation details are provided in Appendix F.

2.3 Related Work

Continual Post-training has become an increasingly important paradigm for extending LLMs beyond their original pretraining distribution, including domain adaptation [26, 27], capability acquisition [28, 29], and behavior alignment over sequential stages [30, 31]. Existing approaches largely follow four lines. Sequential fine-tuning directly adapts the model stage by stage, but is highly prone to forgetting under heterogeneous task sequences [32, 33]. Replay-based methods mitigate forgetting by revisiting historical data [34, 35], while regularization-based methods constrain update drift to preserve prior knowledge [36, 15]. Model merging that combines task-specific adaptations offers a plug-in workflow, but struggles to resolve cross-task interference [18, 37]. However, most existing methods emphasize preserving prior performance during sequential updates while offering limited guidance on the task-compatibility conditions under which sequential interactions should be encouraged or suppressed. Our work addresses this gap through a task-compatibility perspective.

Continual Model Merging provides a data-efficient alternative to standard sequential adaptation by composing task-specific parameter updates in weight space [38, 39, 40]. Recent work studies sequential settings in which models arrive incrementally over time [41, 42, 43], including projection-based sequential merging [44], stability-based methods based on null-space filtering or test-time gating [45, 46], resource-constrained online merging of adapters [47], and broader hybrid frameworks that combine continual learning and model merging [48]. Our method is instantiated as a data-free continual merging method, but the broader goal is to study continual post-training through task compatibility and use merging as an mechanism for exploiting the resulting compatibility findings.

Compatibility Metrics and Signals. Recent work studies compatibility via parameter discrepancy [3, 49], gradient alignment [50], and subspace or spectral overlap [24, 51]. *Demystifying Mergeability* [3] shows that subspace overlap and gradient alignment are stable method-agnostic indicators, but these signals remain largely diagnostic. In contrast, we introduce *geometry conflict* as a method-native control signal derived from task-induced covariance geometry, and construct a shared merging metric via Bures–Wasserstein geometry [20] and Gaussian Wasserstein barycenters [52].

3 What Governs Forgetting in Continual Post-Training?

Before introducing GCWM, we first ask what makes a continual post-training step harmful. Across Qwen3 models [53] from 0.6B to 14B and four representative strategies–Seq. SFT, EWC regularization [54], FOREVER replay [35], and AIMMerging [17]–we compare forgetting with update norm, SAR, gradient conflict, and our geometry conflict. Here, retention loss is the positive old-task drop from each task's best previous score, reported in percentage points (pp) when scaled by 100, and ρ_s denotes Spearman rank correlation. The analysis yields four findings: update norm is only a coarse drift baseline; geometry conflict refines SAR-based compatibility; state-relative geometry mismatch best tracks continual forgetting; geometry and gradient conflict reveal complementary failure modes. Extended diagnostics and bootstrap confidence intervals are organized in Appendix G.

3.1 Update Norm Is Insufficient to Explain Forgetting

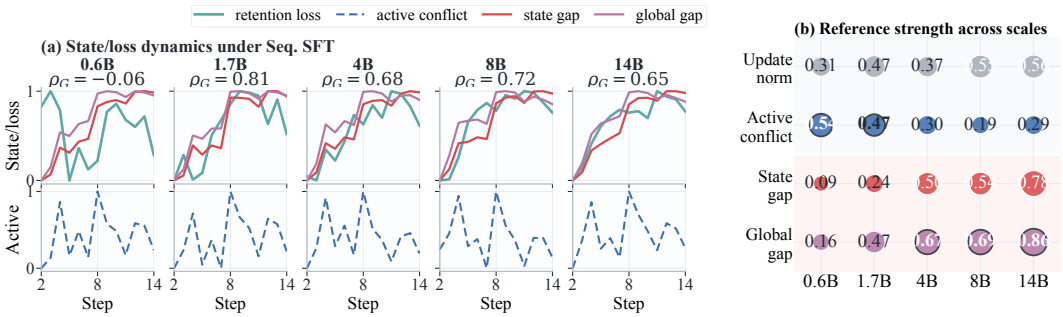
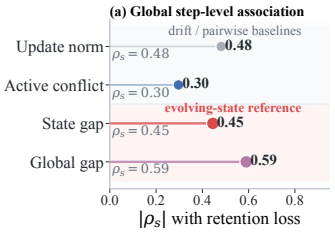


Figure 1: State-relative geometry tracks forgetting across continual steps and scales. Panel (a) shows normalized SFT dynamics within each scale; panel (b) reports $|\rho_s|$ between each signal and loss.

A natural hypothesis is that forgetting is mainly driven by parameter drift: larger updates should induce larger retention loss. We test this by comparing update norm with forgetting, and contrast it with geometry signals that use different reference points. In Figs. 1 and 2, active conflict is the mean pairwise geometry conflict among active task updates, while state and global gaps measure geometry mismatch between active task updates and the evolving model state. Fig. 2(a) shows that update norm has a nontrivial but coarse association with retention loss ($|\rho_s| = 0.48$). State-relative geometry is stronger: the global state-active gap reaches $|\rho_s| = 0.59$, exceeding both update norm and active-pair conflict ($|\rho_s| = 0.30$). The scale breakdown in Fig. 1(b) further shows that this advantage becomes clearer in larger LLMs: the global gap increases from 0.16 at 0.6B to 0.86 at 14B, while update norm remains a weaker drift baseline. Overall, **update norm measures how far the model moves, but not whether the movement remains compatible with task-induced geometries**. Bootstrap confidence intervals and additional step-level rankings are provided in Appendices G.1 and G.3.



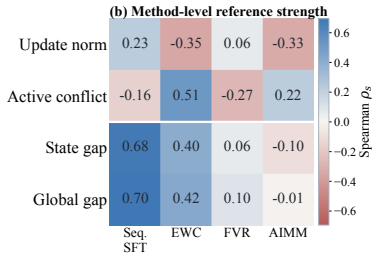


Figure 2: Global and method-level associations. Top: global $|\rho_s|$. Bottom: signed method-level ρ_s; FVR denotes FOREVER.

3.2 Geometry Conflict Refines Subspace Compatibility

Subspace overlap is a natural compatibility proxy: if two updates act on similar directions, they may be easier to integrate. We therefore compare SAR with geometry conflict (Sec. 2.2). As shown in Fig. 3(a), SAR and geometry conflict are related but non-redundant: their global rank association is moderate ($\rho_s = 0.27$), and task pairs with similar SAR can still exhibit very different geometry conflict. SAR captures *where* updates overlap; geometry conflict captures whether their induced covariance geometry is compatible in that shared space.

Pairwise geometry is useful for regime diagnosis, but it is not a standalone predictor of forgetting. In Fig. 3(a–c), SAR percentile ranks task-pair SAR values; GC-drop and GC-forget denote correlations between pairwise

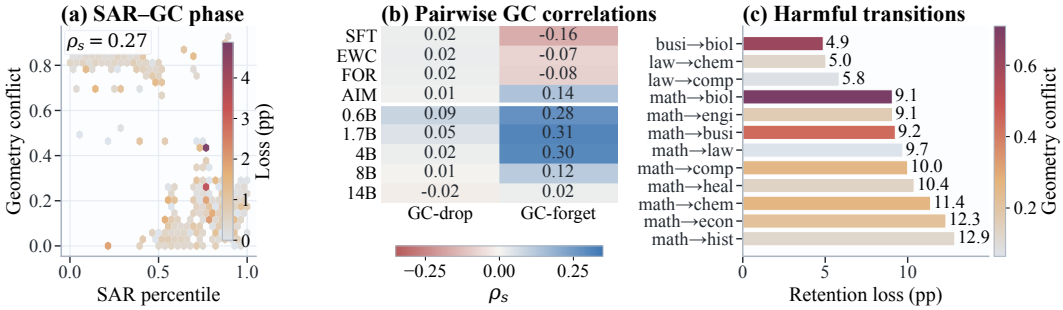


Figure 3: **Pairwise compatibility is informative but insufficient.** (a)–(c) SAR and geometry conflict stratify task-pair transfer regimes, while pairwise conflict alone weakly predicts forgetting. GC-drop is the signed association with the immediate old-task delta; GC-forget measures degradation from each old task's best prior score.

geometry conflict and immediate old-task score change or best-previous forgetting, respectively. Fig. 3(b) shows that GC-drop stays near zero across methods and scales, while GC-forget is scale-sensitive: it is visible on 0.6B–4B ($0.28/0.31/0.30$) but weak on 8B and 14B ($0.12/0.02$). Fig. 3(c) further illustrates this point: large drops, such as Math→History (12.9 pp) and Math→Economics (12.3 pp), do not form a single pairwise-conflict pattern. Thus, pairwise compatibility is informative but insufficient, motivating the state-relative analysis in Sec. 3.3. Overall, **SAR and geometry conflict capture different levels of compatibility**. Pairwise confidence intervals, heatmaps, summaries, and harmful transitions are provided in Appendices G.1 and G.4.

3.3 State-Relative Geometry Conflict Tracks Continual Forgetting

Sec. 3.2 shows that isolated task pairwise compatibility is incomplete. In LLM continual post-training, each incoming update is applied to an evolving model state that already encodes previous updates. The question is whether incoming task geometry remains compatible with the *current* state.

Fig. 1(a) tracks this effect under Seq. SFT. Active-pair conflict fluctuates across steps, while state and global gaps more closely follow the growth of retention loss, especially from 1.7B to 14B. The method-level heatmap in Fig. 2(b) shows the same pattern is strongest under direct sequential updating: state/global signals reach $0.68/0.70$ for Seq. SFT and remain substantial for EWC ($0.40/0.42$), but weaken when replay or merging compresses forgetting variance. This identifies the evolving model state, rather than isolated task pairs, as the relevant reference point for geometry-based forgetting analysis. Full confidence intervals and method-stratified correlations are in Appendices G.2–G.3.

3.4 Geometry and Gradient Conflict Reveal Complementary Failure Modes

Finally, we ask whether geometry conflict simply duplicates gradient conflict. The answer is no. Here, $q/k/v/o$ denote attention projections, $gate/up/down$ denote MLP projections, top-layer share is the fraction of top-ranked conflict layers in each family, min grad-cos is the minimum gradient cosine, and neg-grad ratio is the fraction of negative-cosine pairs. Fig. 4(a) shows a sharp module-level separation: top geometry-conflict layers concentrate in up_proj, $gate_\mathrm{proj}$, v_proj, and $down_\mathrm{proj}$, whereas top negative-gradient layers are dominated by k_proj and q_proj. Together, the four geometry-heavy families account for about 0.89 of top geometry-conflict layers, while query/key projections account for about 0.86 of top negative-gradient layers. Fig. 4(b) further shows that the geometry-conflict locus changes with the update-integration strategy, while negative-gradient conflict remains consistently query/key-centric. Fig. 4(c) complements this module view: the global geometry gap is the strongest forgetting-aligned signal among the plotted predictors, whereas gradient diagnostics are more aligned with old-task mean and overall performance. Thus, **geometry conflict and gradient conflict are complementary diagnostics**. Gradient conflict exposes optimization-level opposition, while geometry conflict captures update-integration mismatch. This distinction is important for GCWM: geometry conflict is not used as a replacement for gradient diagnostics, but as a native signal for controlling how strongly sequential updates should be integrated. Gradient or hybrid gates require task samples to compute gradient conflict, whereas GCWM's data-free gate uses geometry conflict available directly from task updates or checkpoints. Confidence intervals for the geometry and gradient target comparison and decompositions are in Appendices G.1 and G.5.

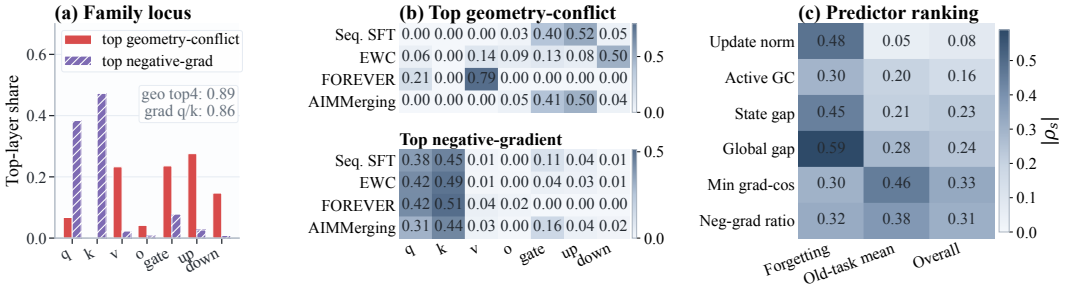


Figure 4: **Geometry and gradient conflict are complementary.** (a)–(c) reveal complementary failure modes: top-layer share is the fraction of top-ranked layers within each projection family, and (c) reports global step-level $|\rho_s|$.

4 Geometry Conflict Wasserstein Merging

We now turn the state-relative geometry findings in Sec. 3 into a data-free update-integration algorithm. **G**eometry-**C**onflict **W**asserstein **M**erging (GCWM) operates on task vectors, estimates layer-wise geometry conflict, constructs a shared Wasserstein metric, and uses a conflict gate to control how strongly geometry-aware correction is applied. At continual step t, GCWM then applies only the incremental change of the update, yielding a compatibility-controlled continual post-training merge. The state and global gaps summarize the evolving model state for analysis and are not directly optimized by GCWM. GCWM uses layer-wise geometry conflict over the complete history-aware active set to construct the shared Wasserstein metric and control the conflict gate.

4.1 Task Geometry and Conflict Gate

GCWM represents each active task update by its layer-wise covariance geometry. For an active update Δ_i and target linear layer ℓ, let $\Delta_i^{(\ell)} \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$. We define

$$C_i^{(\ell)} = (\Delta_i^{(\ell)})^\top \Delta_i^{(\ell)} + \lambda I, \tag{1}$$

which captures the dominant update subspaces and spectral energy while ensuring numerical stability.

To compare multiple active updates in a shared system, GCWM computes a truncated SVD

$$\Delta_i^{(\ell)} \approx U_i^{(\ell)} \Sigma_i^{(\ell)} (V_i^{(\ell)})^\top,$$

retains the principal right-singular directions, and forms

$$Q^{(\ell)} = \operatorname{orth}\Big(\big[V_1^{(\ell)}, V_2^{(\ell)}, \ldots, V_m^{(\ell)} \big] \Big), \tag{2}$$

where m is the number of active task updates. The projected geometry is

$$B_i^{(\ell)} = (Q^{(\ell)})^\top C_i^{(\ell)} Q^{(\ell)}. \tag{3}$$

The operators $\{B_i^{(\ell)}\}_{i=1}^m$ are used for conflict estimation and shared-metric construction.

For two projected geometries, GCWM defines layer-wise geometry conflict by the normalized Bures–Wasserstein discrepancy

$$\gamma_{ij}^{(\ell)} = \frac{d_{\text{B}}^2\Big(B_i^{(\ell)}, B_j^{(\ell)}\Big)}{\operatorname{tr}\Big(B_i^{(\ell)}\Big) + \operatorname{tr}\Big(B_j^{(\ell)}\Big) + \varepsilon}, \qquad d_{\text{B}}^2(A,B) = \operatorname{tr}(A) + \operatorname{tr}(B) - 2\operatorname{tr}\Big(\Big(A^{1/2} B A^{1/2}\Big)^{1/2}\Big), \tag{4}$$

where $\varepsilon > 0$ is a stabilizer. Smaller $\gamma_{ij}^{(\ell)}$ indicates more compatible task-induced geometries. GCWM aggregates pairwise conflicts and converts the result into a layer-wise gate:

$$g^{(\ell)} = \sum_{i<j} w_{ij} \gamma_{ij}^{(\ell)}, \qquad \sum_{i<j} w_{ij} = 1, \tag{5}$$

$$\alpha^{(\ell)} = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min})\, \sigma\big(\kappa(g^{(\ell)} - \tau)\big). \tag{6}$$

Here w_{ij} are normalized task-pair weights, τ is the conflict threshold, and κ controls gate sharpness. Thus, geometry conflict becomes an actionable layer-wise control signal rather than a purely diagnostic score.

4.2 Shared Wasserstein Metric and Gated Merge

Given $\{B_i^{(\ell)}\}_{i=1}^m$, GCWM constructs a shared merging metric through the Gaussian Wasserstein barycenter

$$\bar{B}^{(\ell)} = \arg\min_{B \succeq 0} \sum_{i=1}^{m} \omega_i d_{\mathrm{B}}^2\left(B, B_i^{(\ell)}\right), \qquad \sum_i \omega_i = 1. \tag{7}$$

The barycenter $\bar{B}^{(\ell)}$ defines the local metric in which active updates are aligned before merging.

Let $\hat{\Delta}_i^{(\ell)} = \Delta_i^{(\ell)} Q^{(\ell)}$. GCWM whitens the projected update, applies a base merge operator $\mathcal{M}$, and recolors the result:

$$\tilde{\Delta}_i^{(\ell)} = \hat{\Delta}_i^{(\ell)} \big(\bar{B}^{(\ell)}\big)^{-1/2}, \tag{8}$$

$$\tilde{\Delta}_{\mathrm{geo}}^{(\ell)} = \mathcal{M}\Big(\{\tilde{\Delta}_i^{(\ell)}\}_{i=1}^m; \{\omega_i\}_{i=1}^m\Big),$$

$$\Delta_{\mathrm{geo}}^{(\ell)} = \tilde{\Delta}_{\mathrm{geo}}^{(\ell)} \big(\bar{B}^{(\ell)}\big)^{1/2} \big(Q^{(\ell)}\big)^{\top}. \tag{9}$$

We instantiate $\mathcal{M}$ with weighted WUDI [55]. The geometry-aware branch is then blended with an ungated plain merge:

$$\Delta_{\mathrm{merge}}^{(\ell)} = \alpha^{(\ell)} \Delta_{\mathrm{geo}}^{(\ell)} + \big(1 - \alpha^{(\ell)}\big) \Delta_{\mathrm{plain}}^{(\ell)}. \tag{10}$$

For clarity, Eqs. (8)–(9) present the projected form; the implementation uses the corresponding regularized full-space transform detailed in Appendix C.

4.3 Incremental Continual Update

GCWM is applied incrementally. At step t, let $\mathcal{A}_t$ be the active set of task updates selected by the memory policy, containing the current update and optionally historical updates or the previous merged state. For each target layer, GCWM computes $\Delta_{\mathrm{merge},t}^{(\ell)}$ using Eqs. (4)–(10). Instead of reapplying the full merged update, GCWM applies only its change relative to the previous merged state:

$$\Delta_{\mathrm{inc},t}^{(\ell)} = \Delta_{\mathrm{merge},t}^{(\ell)} - \Delta_{\mathrm{merge},t-1}^{(\ell)}, \qquad \Delta_{\mathrm{merge},0}^{(\ell)} = 0. \tag{11}$$

The model update is $\theta_t^{(\ell)} = \theta_{t-1}^{(\ell)} + \eta_t \Delta_{\mathrm{inc},t}^{(\ell)}$, where η_t is a step coefficient. This rule keeps continual post-training tied to newly induced compatibility-controlled changes.

4.4 Theoretical Support for GCWM

We provide a fixed-step proposal analysis of GCWM relative to the plain merge. This analysis isolates the effect of geometry-aware correction before the incremental differencing in Eq. (11); the implementation applies the change between consecutive merged proposals. Under local smoothness, projected-geometry adequacy, and layer-wise metric-curvature assumptions stated in Appendix D, the relative loss effect of the GCWM correction is controlled by geometry conflict and metric displacement.

Let $\Theta_{\mathrm{plain},t} = \theta_{t-1} + \eta_t \Delta_{\mathrm{plain},t}$, $\Theta_{\mathrm{gcwm},t} = \theta_{t-1} + \eta_t \Delta_{\mathrm{merge},t}$, where $\Delta_{\mathrm{plain},t}$ and $\Delta_{\mathrm{merge},t}$ are the plain and GCWM merge proposals at step t. Let $\widetilde{B}_t^{(\ell)}$ denote the implementation-aligned full-space metric induced by the shared Wasserstein metric.

Theorem 1 (Conflict-Controlled Integration)**.** *Assume $m_t \geq 2$ and $0 \leq \alpha_{\min} \leq \alpha_{\max} \leq 1$. Under the assumptions in Appendix D, the additional loss incurred on a previously acquired task u by the GCWM proposal relative to the plain proposal satisfies*

$$\mathcal{L}_u(\Theta_{\mathrm{gcwm},t}) - \mathcal{L}_u(\Theta_{\mathrm{plain},t}) \leq \eta_t \sum_{\ell} c_{u,t}^{(\ell)} g_t^{(\ell)} + \frac{\eta_t^2}{2} \sum_{\ell} d_{u,t}^{(\ell)} \big\| \Delta_{\mathrm{merge},t}^{(\ell)} - \Delta_{\mathrm{plain},t}^{(\ell)} \big\|_{\widetilde{B}_t^{(\ell)}}^2,$$

where $c_{u,t}^{(\ell)}, d_{u,t}^{(\ell)} \geq 0$ are local constants.

Theorem 1 shows that the relative loss effect of the geometry-aware proposal is bounded by two quantities: the shared geometry conflict $g_t^{(\ell)}$ and the metric displacement from the plain merge. The next result shows how the conflict gate controls this displacement.

Table 1: Domain-continual MMLU-Pro results on Qwen3-1.7B, 8B, and 14B. Scores are accuracies (%). Underlined MTL is a joint-training upper-bound reference; bold marks the best non-MTL result in each block. Data-free update-integration methods are shaded in blue.

Method	Overall	Bio	Bus	Chem	CS	Econ	Eng	Health	Hist	Law	Math	Other	Phil	Phys	Psych
Qwen3-1.7B															
MTL	44.4	65.1	51.6	43.0	47.3	52.3	37.1	41.9	30.2	23.3	53.2	35.8	39.5	46.4	52.9
Seq. SFT	36.8	55.6	39.7	36.3	37.6	48.0	29.0	31.9	24.9	17.7	47.1	31.2	32.1	36.6	47.4
EWC	40.0	58.2	44.6	39.6	40.0	48.6	32.3	35.8	24.1	17.7	55.4	31.7	31.5	42.9	46.6
FOREVER	38.5	58.2	42.2	37.6	41.2	46.3	28.0	35.4	27.3	16.2	50.8	33.1	30.9	40.5	47.4
L&S	41.1	61.4	48.2	39.9	44.0	48.8	34.2	38.8	27.4	20.6	49.8	32.9	36.4	43.2	49.5
AIMMerging	41.8	60.7	49.7	41.4	45.9	50.8	**36.6**	36.9	24.4	18.4	**56.4**	34.4	32.3	42.3	47.5
OPCM	41.7	62.7	49.0	40.4	44.8	49.7	34.5	39.3	27.5	20.5	50.7	33.1	36.8	43.8	50.4
GCWM	**43.5**	**64.9**	**51.0**	**42.2**	**46.6**	**51.7**	36.1	**41.1**	**29.0**	**21.9**	52.7	**34.8**	**38.6**	**45.7**	**52.3**
Qwen3-8B															
MTL	65.3	83.3	70.6	65.9	66.3	74.4	54.3	65.4	58.0	40.2	76.1	58.2	60.7	67.5	73.9
Seq. SFT	55.2	75.2	59.2	53.7	51.7	67.1	48.4	54.4	49.9	27.3	59.1	51.5	51.7	57.0	67.2
EWC	60.4	78.8	66.2	63.0	61.0	71.1	51.8	59.4	49.9	29.9	72.3	52.9	55.3	63.2	68.2
FOREVER	59.6	79.4	63.5	59.6	61.5	69.8	48.9	62.0	50.4	29.9	71.3	54.6	51.9	62.5	68.5
L&S	62.4	79.4	68.3	67.0	**65.4**	70.5	52.6	61.5	51.7	32.3	77.3	55.5	52.1	66.0	67.4
AIMMerging	62.9	78.1	**69.1**	**67.9**	65.1	71.3	**53.0**	62.4	52.0	32.1	**78.7**	55.5	51.9	**67.4**	67.9
OPCM	61.9	78.8	66.8	62.4	62.8	70.5	51.4	61.9	54.9	38.1	72.0	55.1	57.5	63.9	70.0
GCWM	**63.7**	**81.4**	68.9	64.2	64.7	**72.7**	52.7	**63.7**	**56.4**	**38.8**	74.3	**56.6**	**59.1**	65.8	**72.2**
Qwen3-14B															
MTL	68.6	86.2	74.0	72.4	67.8	77.0	56.8	67.3	61.9	39.9	79.4	64.6	61.7	72.2	77.6
Seq. SFT	60.4	79.6	63.4	59.5	63.4	73.3	48.1	61.9	53.8	34.4	68.6	58.2	56.1	58.6	72.6
EWC	65.3	84.2	70.0	67.0	63.4	74.1	52.6	66.3	59.3	33.5	80.9	61.2	55.9	69.1	71.7
FOREVER	66.5	**84.4**	72.1	67.1	69.5	74.6	**56.4**	**66.6**	57.5	35.4	80.5	61.8	58.7	69.7	74.3
L&S	65.6	82.7	70.8	69.2	64.8	73.7	54.1	64.3	59.1	37.7	76.0	61.7	58.9	69.1	74.3
AIMMerging	66.4	83.9	71.7	70.1	65.6	74.7	54.6	65.0	59.7	37.7	77.1	62.4	59.5	70.0	**75.3**
OPCM	66.6	83.7	71.8	70.2	65.8	74.7	55.1	65.3	**60.1**	**38.7**	77.0	**62.7**	**59.9**	70.1	75.3
GCWM	**67.8**	83.8	**73.1**	**72.3**	**72.7**	**76.2**	55.8	66.1	59.6	36.1	**83.4**	61.5	59.5	**73.3**	72.3

Proposition 1 (Compatibility Regimes of Update Integration)**.** *Assume* $0 \le \alpha_{\min} \le \alpha_{\max} \le 1$. *For each layer* ℓ*, let*

$$D_t^{(\ell)} = \left\|\Delta_{\text{geo},t}^{(\ell)} - \Delta_{\text{plain},t}^{(\ell)}\right\|_{\widetilde{B}_t^{(\ell)}}.$$

Then $\left\|\Delta_{\text{merge},t}^{(\ell)} - \Delta_{\text{plain},t}^{(\ell)}\right\|_{\widetilde{B}_t^{(\ell)}} = \alpha_t^{(\ell)} D_t^{(\ell)}, \qquad \left\|\Delta_{\text{merge},t}^{(\ell)} - \Delta_{\text{geo},t}^{(\ell)}\right\|_{\widetilde{B}_t^{(\ell)}} = (1-\alpha_t^{(\ell)}) D_t^{(\ell)}$. *Moreover, Eq.* (6) *implies* $g_t^{(\ell)} \le \tau \Rightarrow \alpha_t^{(\ell)} \le (\alpha_{\min}+\alpha_{\max})/2$*, whereas* $g_t^{(\ell)} \ge \tau \Rightarrow \alpha_t^{(\ell)} \ge (\alpha_{\min}+\alpha_{\max})/2$*. Thus, low-conflict layers receive weaker geometry-aware correction, while high-conflict layers receive stronger correction.*

In summary, Theorem 1 and Proposition 1 show that GCWM is controlled by geometry conflict at both the loss and update levels. Proofs are provided in Appendices D and E.

5 Experiments

We evaluate GCWM as a data-free update-integration method under domain and capability shifts. Our main comparisons focus on data-free merging baselines; sequential, regularized, and replay-based methods are included as reference continual-training pipelines.

5.1 Setup

Models: We use Qwen3 backbones at 0.6B, 1.7B, 4B, 8B, and 14B.
Training data: For domain-continual training, we use [56] and form a 14-task sequence with 1k samples per sub-domain. For capability-continual training, we use 30k math samples from [57] and

Table 2: Capability-continual results on Qwen3-1.7B and 14B. Scores are accuracies or pass@1 (%). MTL is a joint-training reference; Seq. SFT, EWC, and FOREVER are training-pipeline references. Bold marks the best data-free method.

Method	Avg.		GPQA-D.		GSM8K		HumanEval		MATH-500		MBPP		MMLU-Pro	
	1.7B	14B	1.7B	14B	1.7B	14B	1.7B	14B	1.7B	14B	1.7B	14B	1.7B	14B
MTL	57.1	74.6	26.7	33.3	76.1	92.1	61.6	84.2	63.6	87.8	56.8	80.5	57.5	69.8
Seq. SFT	51.9	70.4	18.2	43.4	70.6	95.8	64.0	86.6	64.2	66.2	59.1	63.4	35.3	67.2
EWC	54.7	73.5	24.8	43.4	75.7	95.4	61.6	86.0	67.8	68.0	57.6	78.6	40.5	69.4
FOREVER	58.3	75.9	27.3	54.0	76.7	96.4	62.2	87.2	69.6	69.2	66.9	75.9	47.3	72.8
L&S	52.4	71.3	21.3	38.4	71.0	94.1	62.8	83.7	57.5	76.5	58.4	78.5	43.5	56.8
AIMMerging	53.4	72.2	21.7	38.8	72.4	95.2	64.0	84.7	58.6	77.4	59.5	**79.4**	44.3	57.5
OPCM	56.8	72.9	23.2	38.0	73.0	94.5	65.2	80.5	**67.6**	**79.7**	59.9	78.6	51.9	66.3
GCWM	**58.3**	**74.3**	**26.3**	**39.9**	**79.0**	**95.8**	**67.4**	**86.6**	63.4	78.2	**61.5**	76.7	**52.0**	**68.8**

30k code samples from [58]. More setup details are provided in Appendix H.1 and H.2.
Baselines: Our main baselines are data-free update-integration methods, including Localize-and-Stitch [59], AIMMerging [17], and OPCM [44]. Seq. SFT, EWC [54], and FOREVER [35] are reported as reference continual-training pipelines because they use additional regularization or replay. Non-continual merging like TA [23], TIES [60], DARE [61] are also reported in Appendix H.4.
Benchmarks: Domain-continual performance is evaluated on the 14 MMLU-Pro sub-categories [62]. Capability-continual performance is evaluated on GSM8K [63], MATH-500 [64], MBPP [65], HumanEval [66], GPQA-Diamond [67], and MMLU-Pro [62].
Remark 1: All reported performance scores are averaged over five independent evaluation runs.
Remark 2: The evaluation code we employ strictly adheres to the Qwen3 Technical Report [53]

5.2 Domain-Continual Post-Training

We first evaluate domain-continual post-training on the 14-domain MMLU-Pro sequence. This setting tests whether GCWM can integrate domain-specific updates without accessing training data during the merge stage. Table 1 reports full results on Qwen3-1.7B, 8B, and 14B. MTL is included as a joint-training upper-bound reference, while the main comparison is among data-free update-integration methods. GCWM gives the strongest non-MTL overall performance at all three scales. It improves over the best data-free baseline by $+1.61$, $+0.74$, and $+1.23$ points on Qwen3-1.7B, 8B, and 14B, respectively. The gains are broad rather than driven by a single domain: GCWM improves over AIMMerging on 12/14 domains at 1.7B, 10/14 domains at 8B, and 9/14 domains at 14B. These results support the role of geometry conflict as a practical control signal for data-free continual update integration. Complete results for all model scales are reported in Appendix H.5.

5.3 Capability-Continual Post-Training

We next evaluate capability-continual post-training with sequential math and code updates. Table 2 reports the performance of Qwen3-1.7B and 14B across three key domains: knowledge (GPQA-Diamond, MMLU-Pro), math (GSM8K, MATH-500), and code (HumanEval, MBPP). At 1.7B, GCWM gives the best data-free average (58.3), improving over the strongest data-free baseline (OPCM, 56.8) by +1.52 points and leading on all six benchmarks. At 14B, GCWM remains the strongest data-free method on average (74.3 vs. 72.9 for OPCM) and leads on GPQA-Diamond, GSM8K, HumanEval, and MMLU-Pro. Benchmark-specific gaps to MTL can reflect capability acquisition, data mixing, and update integration, so the GSM8K–MATH-500 contrast does not isolate forgetting or imply that deeper reasoning is intrinsically more forgettable. FOREVER can be higher in some settings because it revisits data through replay and additional sequential optimization; we include it as a replay-based reference, while the primary comparison is among data-free update-integration methods. These results show that geometry-conflict-controlled integration extends beyond domain transfer to heterogeneous capability updates. Full five-scale results are provided in Appendix H.6.

5.4 Ablations and Analysis

We ablate two merge-time components of GCWM: the conflict gate and the shared Wasserstein metric. All variants use the same task experts and evaluation protocol within each setting, differing only in the

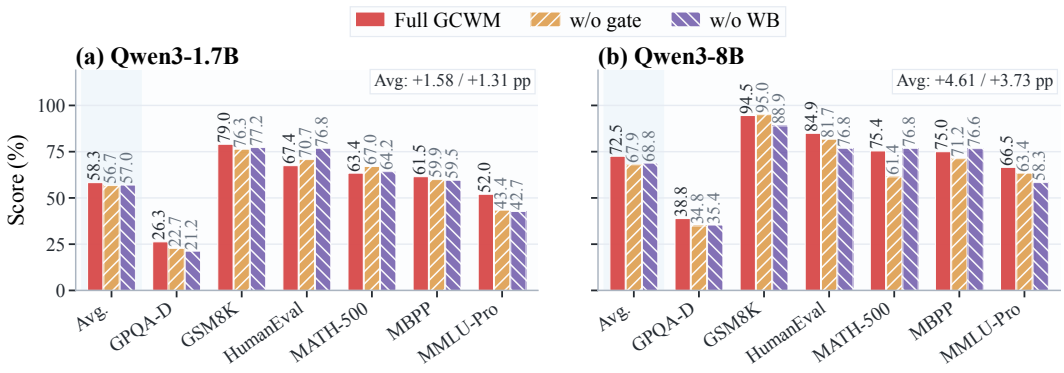


Figure 5: Capability-continual GCWM ablations on Qwen3-1.7B and 8B. All variants use the same task experts within each scale; WB denotes Wasserstein barycenter.

integration rule. The *w/o gate* variant removes conflict-conditioned gating and applies the geometry-aware branch uniformly, while *w/o Wasserstein barycenter* replaces the shared Wasserstein barycenter with a mean covariance metric. Fig. 5 shows that full GCWM improves average performance over w/o gate and w/o Wasserstein barycenter by 1.58 and 1.31 points on Qwen3-1.7B and 4.61 and 3.73 points on Qwen3-8B, respectively. The benchmark-level trade-offs show that compatibility-controlled integration improves aggregate performance without uniformly improving every capability. These results support the gate as a layer-wise control signal and the Wasserstein metric as a shared geometry for data-free update integration. Domain-continual ablations on Qwen3-0.6B and full capability breakdowns are reported in Appendix I.

Remark 3: Appendix K reports Qwen3-8B rank and gate-parameter sensitivity.

Merge-Time Efficiency. We profile offline integration on one GPU at retained rank $r = 16$ (Fig. 6). On Qwen3-8B, increasing active updates from $m = 1$ to 3 raises runtime from 20.6 to 59.9 minutes and peak allocated GPU memory from 4.4 to 11.1 GB. Qwen3-14B requires 51.6–100.8 minutes and 8.4–15.0 GB over two profiled steps, matching 8B at $m = 1, 2$. SVD preparation and merge optimization dominate cost; barycenter computation takes at most 55 seconds. These data-free costs are offline, with no additional inference-time overhead. Appendix J provides more detailed measurements.

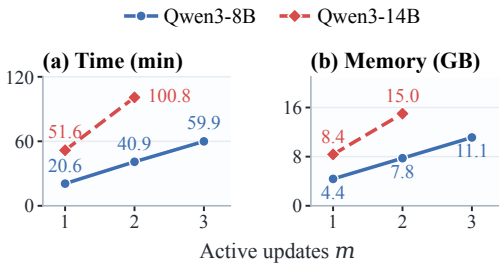


Figure 6: GCWM offline profiling ($r = 16$, one GPU). Qwen3-14B has two measured steps.

6 Conclusion

We studied LLM continual post-training through task geometry, analyzing update norm, subspace alignment, gradient conflict, and geometry conflict across model scales and continual strategies. Our main finding is that forgetting is a state-relative update-integration failure: harmful steps occur when task-induced covariance geometries become incompatible with the geometry of the evolving model state. This explains why raw drift and isolated pairwise compatibility are insufficient, and why geometry conflict serves as both an explanatory signal for forgetting and a control signal for sequential update integration. **G**eometry-**C**onflict **W**asserstein **M**erging (GCWM) operationalizes this insight by constructing a Wasserstein shared metric from task-induced covariance geometry and gating data-free update integration by geometry conflict. Across domain-continual and capability-continual settings, GCWM improves retention and final performance over data-free baselines without replay data.

Acknowledgments

This work was supported by the Hong Kong RGC (TRS: T41-517/25-N; GRF: 15228325), ITC RAISe+ (RAI/24/1/086A), and the Research Institute for Generative Artificial Intelligence at PolyU.

References

[1] 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.

[2] Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip HS Torr, Fahad Shahbaz Khan, and Salman Khan. Llm post-training: A deep dive into reasoning large language models. *arXiv preprint arXiv:2502.21321*, 2025.

[3] Zixuan Ke, Yifei Ming, Xuan-Phi Nguyen, Caiming Xiong, and Shafiq Joty. Demystifying domain-adaptive post-training for financial llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 31021–31047, 2025.

[4] Fei Zhao, Chonggang Lu, Zheyong Xie, Ziyan Liu, Haofu Qian, Jianzhao Huang, Fangcheng Shi, Zijie Meng, Hongcheng Guo, Mingqian He, et al. Redone: Revealing domain-specific llm post-training in social networking services. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 2648–2674, 2025.

[5] Shuo Tang, Xianghe Pang, Zexi Liu, Bohan Tang, Rui Ye, Tian Jin, Xiaowen Dong, Yanfeng Wang, and Siheng Chen. Synthesizing post-training data for llms through multi-agent simulation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23306–23335, 2025.

[6] Taro Yano, Yoichi Ishibashi, and Masafumi Oyamada. Lamdagent: An autonomous framework for post-training pipeline optimization via llm agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 30066–30083, 2025.

[7] Zelin Tan, Hejia Geng, Xiaohang Yu, Mulei Zhang, Guancheng Wan, Yifan Zhou, Qiang He, Xiangyuan Xue, Heng Zhou, Yutao Fan, et al. Scaling behaviors of llm reinforcement learning post-training: An empirical study in mathematical reasoning. *arXiv preprint arXiv:2509.25300*, 2025.

[8] Hongzhe Du, Weikai Li, Min Cai, Karim Saraipour, Zimin Zhang, Yizhou Sun, Himabindu Lakkaraju, and Shichang Zhang. How post-training reshapes llms: A mechanistic view on knowledge, truthfulness, refusal, and confidence. In *The First Workshop on the Application of LLM Explainability to Reasoning and Planning*, 2025.

[9] Gido M Van de Ven, Nicholas Soures, and Dhireesha Kudithipudi. Continual learning and catastrophic forgetting. *arXiv preprint arXiv:2403.05175*, 2024.

[10] Brandon Shuen Yi Loke, Filippo Quadri, Gabriel Vivanco, Maximilian Casagrande, and Saúl Fenollosa. Overcoming catastrophic forgetting in neural networks. *arXiv preprint arXiv:2507.10485*, 2025.

[11] Zixuan Ke, Haowei Lin, Yijia Shao, Hu Xu, Lei Shu, and Bing Liu. Continual training of language models for few-shot learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10205–10216, 2022.

[12] Zhilin Wang, Yafu Li, Xiaoye Qu, and Yu Cheng. See: Continual fine-tuning with sequential ensemble of experts. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7418–7432, 2025.

[13] Truman Hickok. Scalable strategies for continual learning with replay. *arXiv preprint arXiv:2505.12512*, 2025.

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

[15] Hongjoon Ahn, Sungmin Cha, Donggyu Lee, and Taesup Moon. Uncertainty-based continual learning with adaptive regularization. *Advances in neural information processing systems*, 32, 2019.

[16] Jary Pomponi, Simone Scardapane, Vincenzo Lomonaco, and Aurelio Uncini. Efficient continual learning in neural networks with embedding regularization. *Neurocomputing*, 397:139–148, 2020.

[17] Yujie Feng, Jian Li, Xiaoyu Dong, Pengfei Xu, Xiaohui Zhou, Yujia Zhang, Zexin Lu, Yasha Wang, Alan Zhao, Xu Chu, et al. Aimmerging: Adaptive iterative model merging using training trajectories for language model continual learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 13431–13448, 2025.

[18] Dingkun Zhang, Shuhan Qi, Xinyu Xiao, Kehai Chen, and Xuan Wang. Merge then realign: Simple and effective modality-incremental continual learning for multimodal llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 13159–13175, 2025.

[19] Yuanyi Wang, Yanggan Gu, Yiming Zhang, Qi Zhou, Zhaoyi Yan, Congkai Xie, Xinyao Wang, Jianbo Yuan, and Hongxia Yang. Model merging scaling laws in large language models. *arXiv preprint arXiv:2509.24244*, 2025.

[20] Rajendra Bhatia, Tanvi Jain, and Yongdo Lim. On the bures–wasserstein distance between positive definite matrices. *Expositiones mathematicae*, 37(2):165–191, 2019.

[21] Antonio Andrea Gargiulo, Donato Crisostomi, Maria Sofia Bucarelli, Simone Scardapane, Fabrizio Silvestri, and Emanuele Rodola. Task singular vectors: Reducing task interference in model merging. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18695–18705, 2025.

[22] Zirui Wang and Yulia Tsvetkov. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[23] 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*.

[24] Daniel Marczak, Simone Magistri, Sebastian Cygert, Bartłomiej Twardowski, Bagdanov Andrew D, and Joost van de Weije. No task left behind: Isotropic model merging with common and task-specific subspaces. In *39th International Conference on Machine Learning*. Proceedings of Machine Learning Research (PMLR), 2025.

[25] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33:5824–5836, 2020.

[26] Jon Saad-Falcon, Omar Khattab, Keshav Santhanam, Radu Florian, Martin Franz, Salim Roukos, Avirup Sil, Md Sultan, and Christopher Potts. Udapdr: unsupervised domain adaptation via llm prompting and distillation of rerankers. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 11265–11279, 2023.

[27] Johannes Eschbach-Dymanus, Frank Essenberger, Bianka Buschbeck, and Miriam Exel. Exploring the effectiveness of llm domain adaptation for business it machine translation. In *Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 1)*, pages 610–622, 2024.

[28] Wenpeng Yin, Muhao Chen, Rui Zhang, Ben Zhou, Fei Wang, and Dan Roth. Enhancing llm capabilities beyond scaling up. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 1–10, 2024.

[29] Rachit Bansal, Bidisha Samanta, Siddharth Dalmia, Nitish Gupta, Sriram Ganapathy, Abhishek Bapna, Prateek Jain, and Partha Talukdar. Llm augmented llms: Expanding capabilities through composition. In *The Twelfth International Conference on Learning Representations*, 2024.

[30] Dayu Yang, Fumian Chen, and Hui Fang. Behavior alignment: a new perspective of evaluating llm-based conversational recommendation systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2286–2290, 2024.

[31] Wencai Ye, Mingjie Sun, Shuhang Chen, Wenjin Wu, and Peng Jiang. Align^{3}gr: Unified multi-level alignment for llm-based generative recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 16154–16162, 2026.

[32] Jiabao Ji, Yujian Liu, Yang Zhang, Gaowen Liu, Ramana R Kompella, Sijia Liu, and Shiyu Chang. Reversing the forget-retain objectives: An efficient llm unlearning framework from logit difference. *Advances in Neural Information Processing Systems*, 37:12581–12611, 2024.

[33] Fuli Qiao and Mehrdad Mahdavi. Learn more, but bother less: parameter efficient continual learning. *Advances in Neural Information Processing Systems*, 37:97476–97498, 2024.

[34] Yunan Zhang, Shuoran Jiang, Mengchen Zhao, Yuefeng Li, Yang Fan, Xiangping Wu, and Qingcai Chen. Gere: Towards efficient anti-forgetting in continual learning of llm via general samples replay. *arXiv preprint arXiv:2508.04676*, 2025.

[35] Yujie Feng, Hao Wang, Jian Li, Xu Chu, Zhaolu Kang, Yiran Liu, Yasha Wang, Philip S Yu, and Xiao-Ming Wu. Forever: Forgetting curve-inspired memory replay for language model continual learning. *arXiv preprint arXiv:2601.03938*, 2026.

[36] Yuheng Lu, Bingshuo Qian, Caixia Yuan, Huixing Jiang, and Xiaojie Wang. Controlled low-rank adaptation with subspace regularization for continued training on large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 19165–19181, 2025.

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

[38] Yuanyi Wang, Yanggan Gu, Zihao Wang, Kunxi Li, Yifan Yang, Zhaoyi Yan, Congkai Xie, Jianmin Wu, and Hongxia Yang. Mergepipe: A budget-aware parameter management system for scalable llm merging. *arXiv preprint arXiv:2602.13273*, 2026.

[39] Enneng Yang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, Jie Zhang, and Dacheng Tao. Model merging in llms, mllms, and beyond: Methods, theories, applications, and opportunities. *ACM Computing Surveys*, 58(8):1–41, 2026.

[40] Qi Zhou, Yiming Zhang, Yanggan Gu, Yuanyi Wang, Zhijie Sang, Zhaoyi Yan, Zhen Li, Shengyu Zhang, Fei Wu, and Hongxia Yang. Democratizing ai through model fusion: A comprehensive review and future directions. *Nexus*, 2025.

[41] Mei Li, Yuxiang Lu, Qinyan Dai, Suizhi Huang, Yue Ding, and Hongtao Lu. Became: Bayesian continual learning with adaptive model merging. In *Forty-second International Conference on Machine Learning*.

[42] Doanh C Bui, Ba Hung Ngo, Hoai Luan Pham, Khang Nguyen, Maï K Nguyen, and Yasuhiko Nakashima. Mergeslide: Continual model merging and task-to-class prompt-aligned inference for lifelong learning on whole slide images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4859–4868, 2026.

[43] Qi Zhou, Yiming Zhang, Yanggan Gu, Yuanyi Wang, Zhaoyi Yan, Zhen Li, Chi Yung Chung, and Hongxia Yang. Model fusion for scalable and sustainable artificial intelligence: A review and outlook. *Journal of Modern Power Systems and Clean Energy*, 14(1):37–49, 2026.

[44] Anke Tang, Enneng Yang, Li Shen, Yong Luo, Han Hu, Lefei Zhang, Bo Du, and Dacheng Tao. Merging on the fly without retraining: A sequential approach to scalable continual model merging. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[45] Zihuan Qiu, Yi Xu, Chiyuan He, Fanman Meng, Linfeng Xu, Qingbo Wu, and Hongliang Li. Mingle: Mixture of null-space gated low-rank experts for test-time continual model merging. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

[46] Zihuan Qiu, Lei Wang, Yang Cao, Runtong Zhang, Bing Su, Yi Xu, Fanman Meng, Linfeng Xu, Qingbo Wu, and Hongliang Li. Null-space filtering for data-free continual model merging: Preserving transparency, promoting fidelity. *arXiv preprint arXiv:2509.21413*, 2025.

[47] Donald Shenaj, Ondrej Bohdal, Taha Ceritli, Mete Ozay, Pietro Zanuttigh, and Umberto Michieli. K-merge: Online continual merging of adapters for on-device large language models. *arXiv preprint arXiv:2510.13537*, 2025.

[48] Hoang Phan, Sungmin Cha, Tung Lam Tran, and Qi Lei. Toward a holistic approach to continual model merging. *arXiv preprint arXiv:2509.23592*, 2025.

[49] Zhikang Chen, Sen Cui, Deheng Ye, Min Zhang, Gang Niu, Yu Zhang, Masashi Sugiyama, and Tingting Zhu. From coefficients to directions: Rethinking model merging with directional alignment. *arXiv preprint arXiv:2512.00391*, 2025.

[50] Yongxian Wei, Anke Tang, Li Shen, Zixuan Hu, Chun Yuan, and Xiaochun Cao. Modeling multi-task model merging as adaptive projective gradient descent. In *International Conference on Machine Learning*, pages 66178–66193. PMLR, 2025.

[51] Derek Tam, Mohit Bansal, and Colin Raffel. Merging by matching models in task parameter subspaces. *Transactions on Machine Learning Research*.

[52] Pedro C Álvarez-Esteban, E Del Barrio, JA Cuesta-Albertos, and C Matrán. A fixed-point approach to barycenters in wasserstein space. *Journal of Mathematical Analysis and Applications*, 441(2):744–762, 2016.

[53] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[54] 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.

[55] Runxi Cheng, Feng Xiong, Yongxian Wei, Wanyun Zhu, and Chun Yuan. Whoever started the interference should end it: Guiding data-free model merging via task vectors. In *International Conference on Machine Learning*, pages 10121–10143. PMLR, 2025.

[56] UW-Madison-Lee-Lab. Mmlu-pro-cot-train-labeled. `https://huggingface.co/datasets/UW-Madison-Lee-Lab/MMLU-Pro-CoT-Train-Labeled`, 2025. Hugging Face dataset.

[57] NVIDIA. Nemotron-post-training-dataset-v1. `https://huggingface.co/datasets/nvidia/Nemotron-Post-Training-Dataset-v1`, 2025. Hugging Face dataset.

[58] Tianyu Zheng, Ge Zhang, Tianhao Shen, Xueling Liu, Bill Yuchen Lin, Jie Fu, Wenhu Chen, and Xiang Yue. Opencodeinterpreter: Integrating code generation with execution and refinement. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12834–12859, 2024.

[59] Yifei He, Yuzheng Hu, Yong Lin, Tong Zhang, and Han Zhao. Localize-and-stitch: Efficient model merging via sparse task arithmetic. *Transactions on Machine Learning Research*, 2024, 2024.

[60] Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *Advances in neural information processing systems*, 36:7093–7115, 2023.

[61] Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*, 2024.

[62] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

[63] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[64] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

[65] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[66] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[67] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

[68] Yuanyi Wang, Zhaoyi Yan, Yiming Zhang, Qi Zhou, Yanggan Gu, Fei Wu, and Hongxia Yang. Infigfusion: Graph-on-logits distillation via efficient gromov-wasserstein for model fusion. *arXiv preprint arXiv:2505.13893*, 2025.

[69] Yanggan Gu, Yuanyi Wang, Zhaoyi Yan, Yiming Zhang, Qi Zhou, Fei Wu, and Hongxia Yang. Infifpo: Implicit model fusion via preference optimization in large language models. *arXiv preprint arXiv:2505.13878*, 2025.

[70] Yifan Yang, Jinjia Li, Kunxi Li, Puhao Zheng, Yuanyi Wang, Zheyan Qu, Yang Yu, Jianmin Wu, Ming Li, and Hongxia Yang. Inficoevalchain: A blockchain-based decentralized framework for collaborative llm evaluation. *arXiv preprint arXiv:2602.08229*, 2026.

[71] Guodong Du, Junlin Lee, Jing Li, Runhua Jiang, Yifei Guo, Shuyang Yu, Hanting Liu, Sim Kuan Goh, Ho-Kin Tang, Daojing He, and Min Zhang. Parameter competition balancing for model merging. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[72] Guodong Du, Jing Li, Hanting Liu, Runhua Jiang, Shuyang Yu, Yifei Guo, Sim Kuan Goh, and Ho-Kin Tang. Knowledge fusion by evolving weights of language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

[73] Guodong Du, Zitao Fang, Jing Li, Junlin Li, Runhua Jiang, Shuyang Yu, Yifei Guo, Yangneng Chen, Sim Kuan Goh, Ho-Kin Tang, Daojing He, Honghai Liu, and Min Zhang. Neural parameter search for slimmer fine-tuned models and better transfer. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025.

[74] Guodong Du, Zhuo Li, Xuanning Zhou, Junlin Li, Zesheng Shi, Wanyu Lin, Ho-Kin Tang, Xiucheng Li, Fangming Liu, Wenya Wang, Min Zhang, and Jing Li. Knowledge fusion of large language models via modular skillpacks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026.

[75] Guodong Du and Wanyu Lin. Dynamic model merging made slim. In *Advances in Neural Information Processing Systems*, 2026.

A Limitations

Our analysis and experiments focus on Qwen3-scale open LLMs and on domain and capability continual post-training tasks built from public reasoning, knowledge, math, and code benchmarks. Although the state-relative geometry signal is consistent across scales and methods, it should be viewed as an explanatory and control signal rather than a proof of causal necessity for all forms of forgetting. GCWM is data-free at merge time and is therefore most relevant when historical data are unavailable or replay is undesirable; replay-based training can still be stronger when it is allowed to repeatedly revisit past data. The method also assumes access to task-specific updates or checkpoints and incurs additional offline compute cost for geometry construction and Wasserstein metric computation, though this cost does not affect inference.

B Discussion and Broader impacts

GCWM may make continual LLM adaptation more practical by reducing reliance on replay data and by providing diagnostics for harmful update integration. At the same time, easier post-training can also be used to adapt models toward unsafe or misleading behaviors if deployed without appropriate safety evaluation.

GCWM operates in parameter-update space and is designed for data-free continual update integration. This differs from recent distribution- [68] or preference-level [69] fusion methods, which primarily target cross-model knowledge transfer rather than state-relative continual forgetting.

This work does not introduce new safety guarantees; model developers should combine compatibility-controlled merging with standard data governance, red-teaming, and downstream safety checks before deployment.

C Implementation Details of Algorithm

This section summarizes the implementation-aligned version of GCWM. All task updates are constructed relative to the same pretrained model,

$$\Delta_i = \theta_i - \theta_{\mathrm{pre}}.$$

At continual step t, GCWM forms an active set of updates according to a memory policy: either a history-aware policy that retains previous task updates, or an anchor-based policy that merges the current task against the previously merged state. For each target layer, GCWM then computes a shared task geometry, estimates geometry conflict, constructs a shared Wasserstein metric, performs geometry-aware merging, and applies only the incremental change relative to the previous merged state.

For clarity, the main text presents a projected formulation of whitening and recoloring. The implementation uses the corresponding regularized full-space transform, which preserves the projected geometry while regularizing the orthogonal complement.

The full continual update and merge process is summarized in Algorithm 1.

D Proof of Theorem 1

We prove a relative form of Theorem 1, comparing GCWM against the plain merge at the same continual step. This form directly captures the additional effect introduced by the geometry-aware correction. For simplicity, we state the result for the *projected shared-metric branch* of GCWM, which is the main analytical form studied in the paper; the dense-local implementation variant is treated as an efficiency-oriented special case.

Notation. At continual step t, define

$$\Theta_{\mathrm{plain},t} := \theta_{t-1} + \eta_t \Delta_{\mathrm{plain},t}, \qquad \Theta_{\mathrm{GCWM},t} := \theta_{t-1} + \eta_t \Delta_{\mathrm{merge},t},$$

where $\eta_t > 0$ is the outer step coefficient, $\Delta_{\mathrm{plain},t}$ is the ungated merge, and $\Delta_{\mathrm{merge},t}$ is the GCWM update. For each target layer ℓ, define the GCWM correction relative to the plain merge as

$$\delta_t^{(\ell)} := \Delta_{\mathrm{merge},t}^{(\ell)} - \Delta_{\mathrm{plain},t}^{(\ell)}. \tag{12}$$

Algorithm 1: Implementation-aligned GCWM at continual step t

Input: pretrained parameters θ_{pre}, expert models $\{\theta_i\}$, previous merged state $\Delta_{\mathrm{merge},t-1}$, memory policy, task weights $\{\omega_i\}$, hyperparameters $(r, \rho, \lambda, \tau, \kappa, \alpha_{\min}, \alpha_{\max}, \eta_t)$.
Output: updated model parameters θ_t.
Construct task vectors relative to the base model: $\Delta_i = \theta_i - \theta_{\mathrm{pre}}$
Build the active set $\mathcal{A}_t = \{\Delta_i\}_{i=1}^m$ according to the memory policy: history-aware mode uses historical task vectors (or a recent subset) plus the current task; anchor-based mode uses $\Delta_{\mathrm{merge},t-1}$ and the current task
for *each target layer* ℓ **do**
  Stack active layer updates $V^{(\ell)} = \{\Delta_i^{(\ell)}\}_{i=1}^m$
  Shared metric construction:
  if *dense local metric is used at layer* ℓ **then**
    Compute $\Sigma_i^{(\ell)} = (\Delta_i^{(\ell)})^\top \Delta_i^{(\ell)} + \lambda I$, set $\Sigma_{\mathrm{shared}}^{(\ell)} = \sum_i \omega_i \Sigma_i^{(\ell)}$, and compute the layer conflict score from normalized pairwise Bures distances between $\{\Sigma_i^{(\ell)}\}$
  end
  else
    For each $\Delta_i^{(\ell)}$, compute a truncated right SVD and retain $(V_i^{(\ell)}, S_i^{(\ell)})$
    Form $Q^{(\ell)} = \mathrm{orth}\Big([V_1^{(\ell)}, \ldots, V_m^{(\ell)}]\Big)$
    Construct $B_i^{(\ell)} = (Q^{(\ell)})^\top V_i^{(\ell)} \mathrm{diag}\big((S_i^{(\ell)})^2\big)(V_i^{(\ell)})^\top Q^{(\ell)} + \lambda I$
    if *Wasserstein barycenter metric is used* **then**
      compute $B_{\mathrm{shared}}^{(\ell)}$ as the Gaussian Wasserstein barycenter of $\{B_i^{(\ell)}\}$
    end
    else
      compute $B_{\mathrm{shared}}^{(\ell)} = \sum_i \omega_i B_i^{(\ell)}$
    end
    Compute the layer conflict score from normalized pairwise Bures distances between $\{B_i^{(\ell)}\}$
  end
  Gate computation:

$$\alpha^{(\ell)} = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min})\, \sigma\big(\kappa(g^{(\ell)} - \tau)\big).$$

  Merge branch selection:
  if $\alpha^{(\ell)}$ *is below the skip threshold* **then**
    Set $\Delta_{\mathrm{merge},t}^{(\ell)} = \Delta_{\mathrm{plain}}^{(\ell)}$, where $\Delta_{\mathrm{plain}}^{(\ell)}$ is the plain merge operator on $V^{(\ell)}$
  end
  else
    Whiten the active updates under the shared metric:
      dense branch: $\tilde{\Delta}_i^{(\ell)} = \Delta_i^{(\ell)} (\Sigma_{\mathrm{shared}}^{(\ell)})^{-1/2}$
      projected branch: $\tilde{\Delta}_i^{(\ell)} = \mathcal{T}^-_{Q^{(\ell)}, B_{\mathrm{shared}}^{(\ell)}, \lambda}(\Delta_i^{(\ell)})$
    Apply merge operator to $\{\tilde{\Delta}_i^{(\ell)}\}$, obtaining $\tilde{\Delta}_{\mathrm{geo}}^{(\ell)}$
    Recolor the geometry-aware merge:
      dense branch: $\Delta_{\mathrm{geo}}^{(\ell)} = \tilde{\Delta}_{\mathrm{geo}}^{(\ell)} (\Sigma_{\mathrm{shared}}^{(\ell)})^{1/2}$
      projected branch: $\Delta_{\mathrm{geo}}^{(\ell)} = \mathcal{T}^+_{Q^{(\ell)}, B_{\mathrm{shared}}^{(\ell)}, \lambda}(\tilde{\Delta}_{\mathrm{geo}}^{(\ell)})$
    if $\alpha^{(\ell)}$ *is above* 1 *minus the skip threshold* **then**
      Set $\Delta_{\mathrm{merge},t}^{(\ell)} = \Delta_{\mathrm{geo}}^{(\ell)}$
    end
    else
      Compute $\Delta_{\mathrm{plain}}^{(\ell)}$ and blend

$$\Delta_{\mathrm{merge},t}^{(\ell)} = \alpha^{(\ell)} \Delta_{\mathrm{geo}}^{(\ell)} + (1 - \alpha^{(\ell)}) \Delta_{\mathrm{plain}}^{(\ell)}.$$

    end
  end
end
Set $\Delta_{\mathrm{inc},t}^{(\ell)} = \Delta_{\mathrm{merge},t}^{(\ell)} - \Delta_{\mathrm{merge},t-1}^{(\ell)}$ with $\Delta_{\mathrm{merge},0}^{(\ell)} = 0$, and update $\theta_t^{(\ell)} = \theta_{t-1}^{(\ell)} + \eta_t \, \Delta_{\mathrm{inc},t}^{(\ell)}$
return θ_t

By the GCWM blending rule in Eq. (10),

$$\delta_t^{(\ell)} = \alpha_t^{(\ell)} \Big(\Delta_{\mathrm{geo},t}^{(\ell)} - \Delta_{\mathrm{plain},t}^{(\ell)} \Big), \tag{13}$$

where $\alpha_t^{(\ell)} \in [\alpha_{\min}, \alpha_{\max}]$ is the layer-wise gate.

For a layer-wise collection $X = \{X^{(\ell)}\}_{\ell=1}^{L}$, define the Frobenius inner product

$$\langle X, Y \rangle := \sum_{\ell=1}^{L} \mathrm{tr}\big((X^{(\ell)})^\top Y^{(\ell)}\big).$$

For a positive semidefinite matrix M, define the metric-induced norm

$$\|X\|_M^2 := \mathrm{tr}(XMX^\top). \tag{14}$$

Implementation-aligned shared metric. For each layer ℓ and step t, let $Q_t^{(\ell)} \in \mathbb{R}^{d_{\mathrm{in}} \times r_t^{(\ell)}}$ be the shared basis and $\bar{B}_t^{(\ell)} \in \mathbb{R}^{r_t^{(\ell)} \times r_t^{(\ell)}}$ the projected shared metric. To align the theorem with the implementation, define the corresponding full-space regularized metric

$$\widetilde{B}_t^{(\ell)} := Q_t^{(\ell)} \bar{B}_t^{(\ell)} \big(Q_t^{(\ell)}\big)^\top + \lambda\Big(I - Q_t^{(\ell)} \big(Q_t^{(\ell)}\big)^\top\Big), \tag{15}$$

where $\lambda > 0$ is the regularization parameter and I is the identity matrix of size $d_{\mathrm{in}} \times d_{\mathrm{in}}$. The norm used in the proof is

$$\|X\|_{\widetilde{B}_t^{(\ell)}}^2 = \mathrm{tr}\big(X \widetilde{B}_t^{(\ell)} X^\top\big).$$

This is the metric induced by the full-space whitening/recoloring operators used in the implementation.

Center mismatch quantity. Let $\{B_{i,t}^{(\ell)}\}_{i=1}^{m_t}$ denote the projected task geometries at layer ℓ and step t, and let $\bar{B}_t^{(\ell)}$ be their shared Wasserstein barycenter. Define the weighted center mismatch

$$r_t^{(\ell)} := \sum_{i=1}^{m_t} \omega_{i,t}\, d_{\mathrm{B}}^2\big(B_{i,t}^{(\ell)}, \bar{B}_t^{(\ell)}\big), \tag{16}$$

where $\omega_{i,t} \geq 0$, $\sum_i \omega_{i,t} = 1$, and $d_{\mathrm{B}}(\cdot,\cdot)$ is the Bures distance from [20] This quantity measures how far the active task geometries lie from the shared center.

We use the following noraml assumptions.

Assumption 1 (Local smoothness)**.** *For every previously acquired task u, the loss $\mathcal{L}_u$ is twice continuously differentiable in a neighborhood of the line segment*

$$\big\{\Theta_{\mathrm{plain},t} + s(\Theta_{\mathrm{GCWM},t} - \Theta_{\mathrm{plain},t}) : s \in [0,1]\big\}.$$

Assumption 2 (Projected-geometry adequacy)**.** *For every previously acquired task u, target layer ℓ, and step t, there exists a nonnegative constant $a_{u,t}^{(\ell)}$ such that*

$$\big|\langle \nabla_\ell \mathcal{L}_u(\Theta_{\mathrm{plain},t}), \delta_t^{(\ell)} \rangle\big| \;\leq\; a_{u,t}^{(\ell)}\, r_t^{(\ell)}. \tag{17}$$

This assumption formalizes that the projected geometry captures the dominant first-order directions relevant to cross-task interference.

Assumption 3 (Layer-separable metric curvature bound)**.** *For every previously acquired task u, step t, and $s \in [0,1]$, there exist nonnegative constants $d_{u,t}^{(\ell)}$ such that*

$$\Big\langle \delta_t,\, \nabla^2 \mathcal{L}_u\big(\Theta_{\mathrm{plain},t} + s\eta_t \delta_t\big)\, \delta_t \Big\rangle \;\leq\; \sum_{\ell=1}^{L} d_{u,t}^{(\ell)}\, \|\delta_t^{(\ell)}\|_{\widetilde{B}_t^{(\ell)}}^2. \tag{18}$$

This assumption allows cross-layer couplings in the full Hessian to be absorbed into layer-wise constants while measuring displacement in the implementation-aligned full-space metric $\widetilde{B}_t^{(\ell)}$.

The next lemma shows that the center mismatch $r_t^{(\ell)}$ can be controlled by the pairwise geometry conflict $g_t^{(\ell)}$ used by GCWM.

Lemma 1 (Center mismatch is controlled by pairwise geometry conflict)**.** *Fix a layer ℓ and step t, and assume $m_t \geq 2$. Assume that the pairwise weights in Eq.* (5) *are chosen as*

$$w_{ij,t} = \frac{\omega_{i,t}\omega_{j,t}}{Z_t}, \qquad Z_t := \sum_{1\leq i<j\leq m_t} \omega_{i,t}\omega_{j,t}.$$

Let

$$M_t^{(\ell)} := \max_{1\leq i<j\leq m_t} \Big(\mathrm{tr}(B_{i,t}^{(\ell)}) + \mathrm{tr}(B_{j,t}^{(\ell)}) + \varepsilon\Big). \tag{19}$$

Then

$$r_t^{(\ell)} \;\leq\; M_t^{(\ell)}\, g_t^{(\ell)}. \tag{20}$$

When $m_t = 1$, both $r_t^{(\ell)}$ and $g_t^{(\ell)}$ vanish, so the bound is trivial.

Proof. For fixed ℓ, t, define the Fréchet functional

$$F(B) := \sum_{i=1}^{m_t} \omega_{i,t}\, d_{\mathrm{B}}^2(B_{i,t}^{(\ell)}, B).$$

By construction, $\bar{B}_t^{(\ell)}$ is a minimizer of F. Therefore, for every $j \in \{1, \ldots, m_t\}$,

$$F(\bar{B}_t^{(\ell)}) \leq F(B_{j,t}^{(\ell)}).$$

Multiplying both sides by $\omega_{j,t}$ and summing over j yields

$$\begin{aligned} r_t^{(\ell)} = F(\bar{B}_t^{(\ell)}) &\leq \sum_{j=1}^{m_t} \omega_{j,t} \sum_{i=1}^{m_t} \omega_{i,t}\, d_{\mathrm{B}}^2(B_{i,t}^{(\ell)}, B_{j,t}^{(\ell)}) \\ &= \sum_{i=1}^{m_t}\sum_{j=1}^{m_t} \omega_{i,t}\omega_{j,t}\, d_{\mathrm{B}}^2(B_{i,t}^{(\ell)}, B_{j,t}^{(\ell)}). \end{aligned} \tag{21}$$

Because the diagonal terms vanish, this becomes

$$r_t^{(\ell)} \leq 2 \sum_{1\leq i<j\leq m_t} \omega_{i,t}\omega_{j,t}\, d_{\mathrm{B}}^2(B_{i,t}^{(\ell)}, B_{j,t}^{(\ell)}). \tag{22}$$

Next, by the definition of the normalized pairwise conflict in Eq. (4),

$$d_{\mathrm{B}}^2(B_{i,t}^{(\ell)}, B_{j,t}^{(\ell)}) = \gamma_{ij,t}^{(\ell)} \Big(\mathrm{tr}(B_{i,t}^{(\ell)}) + \mathrm{tr}(B_{j,t}^{(\ell)}) + \varepsilon\Big) \leq M_t^{(\ell)} \gamma_{ij,t}^{(\ell)}.$$

Substituting this bound into Eq. (22) gives

$$r_t^{(\ell)} \leq 2M_t^{(\ell)} \sum_{1\leq i<j\leq m_t} \omega_{i,t}\omega_{j,t}\, \gamma_{ij,t}^{(\ell)}. \tag{23}$$

Using the definition of $w_{ij,t}$, we obtain

$$\sum_{1\leq i<j\leq m_t} \omega_{i,t}\omega_{j,t}\, \gamma_{ij,t}^{(\ell)} = Z_t \sum_{1\leq i<j\leq m_t} w_{ij,t}\, \gamma_{ij,t}^{(\ell)} = Z_t\, g_t^{(\ell)}.$$

Hence

$$r_t^{(\ell)} \leq 2M_t^{(\ell)} Z_t\, g_t^{(\ell)}.$$

Finally, since $\sum_i \omega_{i,t} = 1$,

$$Z_t = \sum_{i<j} \omega_{i,t}\omega_{j,t} = \frac{1 - \sum_i \omega_{i,t}^2}{2} \leq \frac{1}{2}.$$

Therefore

$$r_t^{(\ell)} \leq M_t^{(\ell)}\, g_t^{(\ell)},$$

which proves Eq. (20). □

We are now ready to prove the main result.

The case $m_t = 1$ is trivial, since then the pairwise geometry conflict vanishes and GCWM reduces to the plain merge. We therefore state the theorem for $m_t \geq 2$.

Theorem (Conflict-Controlled Integration)**.** *Assume $m_t \geq 2$ for every analyzed layer ℓ. Under Assumptions 1, 2, and 3, and with pairwise weights chosen as in Lemma 1, the additional loss incurred on a previously acquired task u by GCWM relative to the plain merge satisfies*

$$\mathcal{L}_u(\Theta_{\mathrm{GCWM},t}) - \mathcal{L}_u(\Theta_{\mathrm{plain},t}) \;\leq\; \eta_t \sum_{\ell=1}^{L} c_{u,t}^{(\ell)}\, g_t^{(\ell)} \;+\; \frac{\eta_t^2}{2} \sum_{\ell=1}^{L} d_{u,t}^{(\ell)} \big\| \Delta_{\mathrm{merge},t}^{(\ell)} - \Delta_{\mathrm{plain},t}^{(\ell)} \big\|_{\widetilde{B}_t^{(\ell)}}^2, \quad (24)$$

where

$$c_{u,t}^{(\ell)} := a_{u,t}^{(\ell)} M_t^{(\ell)}.$$

Hence, the additional loss induced by GCWM relative to the plain merge is controlled by shared geometry conflict and gated merge displacement.

Proof. By Assumption 1, the exact second-order Taylor formula in integral form yields

$$\begin{aligned} \mathcal{L}_u(\Theta_{\mathrm{GCWM},t}) - \mathcal{L}_u(\Theta_{\mathrm{plain},t}) &= \Big\langle \nabla \mathcal{L}_u(\Theta_{\mathrm{plain},t}), \Theta_{\mathrm{GCWM},t} - \Theta_{\mathrm{plain},t} \Big\rangle \\ &+ \int_0^1 (1-s) \Big\langle \Theta_{\mathrm{GCWM},t} - \Theta_{\mathrm{plain},t},\, \nabla^2 \mathcal{L}_u\big(\Theta_{\mathrm{plain},t} + s(\Theta_{\mathrm{GCWM},t} - \Theta_{\mathrm{plain},t})\big) \\ &\qquad (\Theta_{\mathrm{GCWM},t} - \Theta_{\mathrm{plain},t}) \Big\rangle ds. \end{aligned} \quad (25)$$

Since

$$\Theta_{\mathrm{GCWM},t} - \Theta_{\mathrm{plain},t} = \eta_t \delta_t,$$

Eq. (25) becomes

$$\begin{aligned} \mathcal{L}_u(\Theta_{\mathrm{GCWM},t}) - \mathcal{L}_u(\Theta_{\mathrm{plain},t}) &= \eta_t \langle \nabla \mathcal{L}_u(\Theta_{\mathrm{plain},t}), \delta_t \rangle \\ &+ \eta_t^2 \int_0^1 (1-s) \Big\langle \delta_t,\, \nabla^2 \mathcal{L}_u\big(\Theta_{\mathrm{plain},t} + s\eta_t \delta_t\big)\, \delta_t \Big\rangle ds. \end{aligned} \quad (26)$$

We first bound the linear term. By layer-wise decomposition,

$$\langle \nabla \mathcal{L}_u(\Theta_{\mathrm{plain},t}), \delta_t \rangle = \sum_{\ell=1}^{L} \langle \nabla_\ell \mathcal{L}_u(\Theta_{\mathrm{plain},t}), \delta_t^{(\ell)} \rangle. \quad (27)$$

Applying Assumption 2,

$$\big| \langle \nabla \mathcal{L}_u(\Theta_{\mathrm{plain},t}), \delta_t \rangle \big| \leq \sum_{\ell=1}^{L} a_{u,t}^{(\ell)}\, r_t^{(\ell)}. \quad (28)$$

By Lemma 1,

$$r_t^{(\ell)} \leq M_t^{(\ell)} g_t^{(\ell)}.$$

Hence

$$\big| \langle \nabla \mathcal{L}_u(\Theta_{\mathrm{plain},t}), \delta_t \rangle \big| \leq \sum_{\ell=1}^{L} a_{u,t}^{(\ell)} M_t^{(\ell)} g_t^{(\ell)} = \sum_{\ell=1}^{L} c_{u,t}^{(\ell)} g_t^{(\ell)}. \quad (29)$$

Therefore

$$\eta_t \langle \nabla \mathcal{L}_u(\Theta_{\mathrm{plain},t}), \delta_t \rangle \leq \eta_t \sum_{\ell=1}^{L} c_{u,t}^{(\ell)} g_t^{(\ell)}. \quad (30)$$

We next bound the second-order term. By Assumption 3, for every $s \in [0,1]$,

$$\Big\langle \delta_t,\, \nabla^2 \mathcal{L}_u\big(\Theta_{\mathrm{plain},t} + s\eta_t \delta_t\big)\, \delta_t \Big\rangle \leq \sum_{\ell=1}^{L} d_{u,t}^{(\ell)}\, \|\delta_t^{(\ell)}\|_{\widetilde{B}_t^{(\ell)}}^2. \quad (31)$$

Substituting Eq. (31) into the integral term in Eq. (26) yields

$$\begin{aligned}
&\eta_t^2 \int_0^1 (1-s) \left\langle \delta_t,\, \nabla^2 \mathcal{L}_u\big(\Theta_{\text{plain},t} + s\eta_t \delta_t\big)\, \delta_t \right\rangle ds \\
&\le \eta_t^2 \int_0^1 (1-s)\, ds \sum_{\ell=1}^{L} d_{u,t}^{(\ell)} \, \|\delta_t^{(\ell)}\|^2_{\widetilde{B}_t^{(\ell)}} \\
&= \frac{\eta_t^2}{2} \sum_{\ell=1}^{L} d_{u,t}^{(\ell)} \, \|\delta_t^{(\ell)}\|^2_{\widetilde{B}_t^{(\ell)}}.
\end{aligned} \tag{32}$$

Combining Eqs. (26), (30), and (32), we obtain

$$\mathcal{L}_u(\Theta_{\text{GCWM},t}) - \mathcal{L}_u(\Theta_{\text{plain},t}) \le \eta_t \sum_{\ell=1}^{L} c_{u,t}^{(\ell)} g_t^{(\ell)} + \frac{\eta_t^2}{2} \sum_{\ell=1}^{L} d_{u,t}^{(\ell)} \, \|\delta_t^{(\ell)}\|^2_{\widetilde{B}_t^{(\ell)}}. \tag{33}$$

Finally, substituting

$$\delta_t^{(\ell)} = \Delta_{\text{merge},t}^{(\ell)} - \Delta_{\text{plain},t}^{(\ell)}$$

from Eq. (12) into Eq. (33) gives Eq. (24). This completes the proof. □

Remark 1 (Gate-scaled displacement)**.** *By Eq.* (13)*,*

$$\|\delta_t^{(\ell)}\|^2_{\widetilde{B}_t^{(\ell)}} = \big(\alpha_t^{(\ell)}\big)^2 \big\|\Delta_{\text{geo},t}^{(\ell)} - \Delta_{\text{plain},t}^{(\ell)}\big\|^2_{\widetilde{B}_t^{(\ell)}}. \tag{34}$$

Thus, the GCWM gate directly scales the second-order displacement term in Theorem D. This is the precise sense in which geometry conflict becomes an actionable control signal.

E Proof of Proposition 1

For convenience, we restate the main definitions. At continual step t and layer ℓ, GCWM forms the merged update

$$\Delta_{\text{merge},t}^{(\ell)} = \alpha_t^{(\ell)} \Delta_{\text{geo},t}^{(\ell)} + \big(1 - \alpha_t^{(\ell)}\big) \Delta_{\text{plain},t}^{(\ell)}, \tag{35}$$

where $\Delta_{\text{geo},t}^{(\ell)}$ is the geometry-aware merge, $\Delta_{\text{plain},t}^{(\ell)}$ is the plain merge, and $\alpha_t^{(\ell)} \in [\alpha_{\min}, \alpha_{\max}]$ is the layer-wise gate. We also define

$$D_t^{(\ell)} := \big\|\Delta_{\text{geo},t}^{(\ell)} - \Delta_{\text{plain},t}^{(\ell)}\big\|_{\widetilde{B}_t^{(\ell)}}, \tag{36}$$

where $\widetilde{B}_t^{(\ell)}$ is the implementation-aligned shared metric defined in Eq. (15).

The gate is given by Eq. (6), namely

$$\alpha_t^{(\ell)} = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min})\, \sigma\big(\kappa(g_t^{(\ell)} - \tau)\big), \tag{37}$$

where $\sigma(z) = 1/(1+e^{-z})$ is the sigmoid function, $\kappa > 0$ is the sharpness parameter, and τ is the conflict threshold.

Proof. We first prove the two norm identities. Subtracting $\Delta_{\text{plain},t}^{(\ell)}$ from both sides of Eq. (35) gives

$$\begin{aligned}
\Delta_{\text{merge},t}^{(\ell)} - \Delta_{\text{plain},t}^{(\ell)} &= \alpha_t^{(\ell)} \Delta_{\text{geo},t}^{(\ell)} + \big(1 - \alpha_t^{(\ell)}\big) \Delta_{\text{plain},t}^{(\ell)} - \Delta_{\text{plain},t}^{(\ell)} \\
&= \alpha_t^{(\ell)} \Big(\Delta_{\text{geo},t}^{(\ell)} - \Delta_{\text{plain},t}^{(\ell)} \Big).
\end{aligned} \tag{38}$$

Taking the norm induced by $\widetilde{B}_t^{(\ell)}$ and using positive homogeneity,

$$\begin{aligned}
\big\|\Delta_{\text{merge},t}^{(\ell)} - \Delta_{\text{plain},t}^{(\ell)}\big\|_{\widetilde{B}_t^{(\ell)}} &= \big\|\alpha_t^{(\ell)} \big(\Delta_{\text{geo},t}^{(\ell)} - \Delta_{\text{plain},t}^{(\ell)}\big)\big\|_{\widetilde{B}_t^{(\ell)}} \\
&= \alpha_t^{(\ell)} \big\|\Delta_{\text{geo},t}^{(\ell)} - \Delta_{\text{plain},t}^{(\ell)}\big\|_{\widetilde{B}_t^{(\ell)}} \\
&= \alpha_t^{(\ell)} D_t^{(\ell)}.
\end{aligned} \tag{39}$$

Similarly, subtracting $\Delta^{(\ell)}_{\mathrm{geo},t}$ from both sides of Eq. (35) gives

$$\begin{aligned}\Delta^{(\ell)}_{\mathrm{merge},t} - \Delta^{(\ell)}_{\mathrm{geo},t} &= \alpha_t^{(\ell)}\Delta^{(\ell)}_{\mathrm{geo},t} + \big(1-\alpha_t^{(\ell)}\big)\Delta^{(\ell)}_{\mathrm{plain},t} - \Delta^{(\ell)}_{\mathrm{geo},t} \\ &= \big(1-\alpha_t^{(\ell)}\big)\Big(\Delta^{(\ell)}_{\mathrm{plain},t} - \Delta^{(\ell)}_{\mathrm{geo},t}\Big). \end{aligned} \tag{40}$$

Taking the same norm, and using the fact that $\|X\|_{\widetilde{B}_t^{(\ell)}} = \|-X\|_{\widetilde{B}_t^{(\ell)}}$,

$$\begin{aligned}\big\|\Delta^{(\ell)}_{\mathrm{merge},t} - \Delta^{(\ell)}_{\mathrm{geo},t}\big\|_{\widetilde{B}_t^{(\ell)}} &= \big(1-\alpha_t^{(\ell)}\big)\big\|\Delta^{(\ell)}_{\mathrm{plain},t} - \Delta^{(\ell)}_{\mathrm{geo},t}\big\|_{\widetilde{B}_t^{(\ell)}} \\ &= \big(1-\alpha_t^{(\ell)}\big)\big\|\Delta^{(\ell)}_{\mathrm{geo},t} - \Delta^{(\ell)}_{\mathrm{plain},t}\big\|_{\widetilde{B}_t^{(\ell)}} \\ &= \big(1-\alpha_t^{(\ell)}\big)D_t^{(\ell)}. \end{aligned} \tag{41}$$

We next prove the threshold characterization of the gate. Since $\kappa > 0$ and the sigmoid function $\sigma(\cdot)$ is monotone increasing with $\sigma(0) = 1/2$, Eq. (37) implies

$$g_t^{(\ell)} \le \tau \quad\Longrightarrow\quad \sigma\big(\kappa(g_t^{(\ell)} - \tau)\big) \le \frac{1}{2},$$

and therefore

$$\begin{aligned}\alpha_t^{(\ell)} &= \alpha_{\min} + (\alpha_{\max} - \alpha_{\min})\,\sigma\big(\kappa(g_t^{(\ell)} - \tau)\big) \\ &\le \alpha_{\min} + \frac{\alpha_{\max} - \alpha_{\min}}{2} = \frac{\alpha_{\min} + \alpha_{\max}}{2}. \end{aligned} \tag{42}$$

Likewise,

$$g_t^{(\ell)} \ge \tau \quad\Longrightarrow\quad \sigma\big(\kappa(g_t^{(\ell)} - \tau)\big) \ge \frac{1}{2},$$

which gives

$$\alpha_t^{(\ell)} \ge \frac{\alpha_{\min} + \alpha_{\max}}{2}. \tag{43}$$

Combining Eqs. (39), (41), (42), and (43) establishes the proposition. In particular, when $g_t^{(\ell)}$ is below the threshold τ, GCWM applies a weaker geometry-aware correction, whereas when $g_t^{(\ell)}$ exceeds τ, the geometry-aware branch receives a larger weight. This establishes the regime characterization claimed in Proposition 1. □

F Analysis Metrics

We use four families of signals to analyze continual post-training dynamics. For a step update Δ_t, we measure parameter drift by the update norm

$$n_t = \Big(\sum_{\ell\in\mathcal{L}} \|\Delta_t^{(\ell)}\|_F^2\Big)^{1/2}.$$

Given the retained right-singular subspace $V_j^{(\ell)}$ of task T_j, we measure subspace overlap by the subspace alignment ratio (SAR)

$$\mathrm{SAR}^{(\ell)}_{i\to j} = \frac{\|\Delta_i^{(\ell)} V_j^{(\ell)}\|_F}{\|\Delta_i^{(\ell)}\|_F + \varepsilon}, \qquad \mathrm{SAR}^{(\ell)}_{ij} = \frac{1}{2}\big(\mathrm{SAR}^{(\ell)}_{i\to j} + \mathrm{SAR}^{(\ell)}_{j\to i}\big).$$

For projected task geometries $B_i^{(\ell)}$ and $B_j^{(\ell)}$, we define geometry conflict by the normalized Bures–Wasserstein discrepancy

$$\gamma^{(\ell)}_{ij} = \frac{d^2_{\mathrm{B}}(B_i^{(\ell)}, B_j^{(\ell)})}{\mathrm{tr}(B_i^{(\ell)}) + \mathrm{tr}(B_j^{(\ell)}) + \varepsilon}.$$

For gradient-based diagnostics, we compute gradient cosine similarity

$$c^{(\ell)}_{ij} = \frac{\langle g_i^{(\ell)}, g_j^{(\ell)}\rangle}{\|g_i^{(\ell)}\|_2 \|g_j^{(\ell)}\|_2 + \varepsilon},$$

and report both mean cosine and the fraction of negative-cosine pairs. State-relative variants replace one task update by the current continual-training state.

Geometry aggregation and gaps. Let $\mathcal{L}$ be the analyzed linear layers, $\mathcal{A}_t = \{\Delta_{i,t}\}_{i=1}^{m_t}$ the active updates, and $S_t = \theta_t - \theta_{\text{pre}}$ the accumulated update of the saved checkpoint after step t. Write $\gamma^{(\ell)}(X, Y; Q)$ for the normalized geometry conflict above evaluated in basis Q; omitting Q denotes the pair-specific union of the retained right-singular subspaces of X and Y. For $m_t \geq 2$, active-pair conflict, state-active conflict, and the state gap are

$$\begin{aligned}
\mathrm{GC}_{\mathrm{active},t} &= \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} \frac{1}{\binom{m_t}{2}} \sum_{i<j} \gamma^{(\ell)}(\Delta_{i,t}, \Delta_{j,t}), \\
\mathrm{GC}_{\mathrm{state},t} &= \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} \frac{1}{m_t} \sum_{i} \gamma^{(\ell)}(S_t, \Delta_{i,t}), \\
\mathrm{Gap}_{\mathrm{state},t} &= \mathrm{GC}_{\mathrm{state},t} - \mathrm{GC}_{\mathrm{active},t}.
\end{aligned}$$

The global gap uses a common union basis for each object set $\mathcal{U}$:

$$\begin{aligned}
Q_{\mathcal{U}}^{(\ell)} &= \mathrm{orth}\Big([V_X^{(\ell)}]_{X \in \mathcal{U}}\Big), \\
\mathrm{GC}_{\mathrm{global}}(\mathcal{U}) &= \frac{1}{|\mathcal{L}|} \sum_{\ell \in \mathcal{L}} \frac{1}{\binom{|\mathcal{U}|}{2}} \sum_{\{X,Y\} \subset \mathcal{U}} \gamma^{(\ell)}(X, Y; Q_{\mathcal{U}}^{(\ell)}), \\
\mathrm{Gap}_{\mathrm{global},t} &= \mathrm{GC}_{\mathrm{global}}(\mathcal{A}_t \cup \{S_t\}) - \mathrm{GC}_{\mathrm{global}}(\mathcal{A}_t).
\end{aligned}$$

Thus, adding the state changes both the object set and the shared basis in the global gap. Both gaps are signed contrasts of averaged geometry conflicts, not additional distance metrics. All means weight analyzed layers and valid pairs uniformly, without weighting by update norm, layer dimension, or retained rank; pairwise means are zero when fewer than two objects are available.

Old-task changes and pairwise correlations. Let $a_{u,t}$ be the score on old task u after step t, and $b_{u,t-1} = \max_{1 \leq s < t} a_{u,s}$ its best previous score. Immediate score change and forgetting from best are

$$\delta a_{u,t} = a_{u,t} - a_{u,t-1}, \qquad f_{u,t} = a_{u,t} - b_{u,t-1}.$$

Negative values indicate degradation; positive values indicate improvement over the corresponding reference. With $\bar{\gamma}_{u,t} = |\mathcal{L}|^{-1} \sum_{\ell \in \mathcal{L}} \gamma^{(\ell)}(\Delta_u, \Delta_t)$, the layer-averaged conflict between old-task and incoming-task updates, we report

$$\text{GC-drop} = \rho_s(\bar{\gamma}_{u,t}, \delta a_{u,t}), \qquad \text{GC-forget} = \rho_s(\bar{\gamma}_{u,t}, f_{u,t}).$$

Each correlation pools valid old-task/incoming-task observations in the indicated global, scale, or method group; steps with no old task are excluded. For the valid old-task set $\mathcal{O}_t$, signed step-level forgetting is $F_t = |\mathcal{O}_t|^{-1} \sum_{u \in \mathcal{O}_t} f_{u,t}$; retention-loss plots use $100 \max(0, -F_t)$ in percentage points. Panels reporting $|\rho_s|$ show association strength without its sign.

G Additional Empirical Analysis for Sec. 3

This appendix mirrors the four empirical findings in Sec. 3: diagnostic dashboards, step-level and state-relative analysis, pairwise compatibility, and module-level geometry–gradient complementarity.

G.1 Statistical Confidence for Sec. 3

To quantify statistical uncertainty in Sec. 3, we report run-cluster bootstrap confidence intervals and permutation-test significance for key Spearman associations. For each statistic, we use 2,000 bootstrap resamples with clusters defined by run (model-size $\times$ method sequence), which preserves within-run temporal dependence across continual steps. We also report two-sided permutation p-values with 3,000 shuffles.

Fig. 7 and Tables 3–7 support the robustness of the main Sec. 3 claims. At the global step level, global state-relative geometry gap remains the strongest forgetting-associated signal ($\rho_s = -0.59$, 95% CI $[-0.74, -0.36]$), stronger than update norm (-0.48, CI $[-0.61, -0.20]$) and active conflict (0.30, CI $[0.00, 0.58]$). By scale, the state/global confidence intervals become more negative and better separated on larger models (4B–14B), while 0.6B intervals are substantially wider. At the

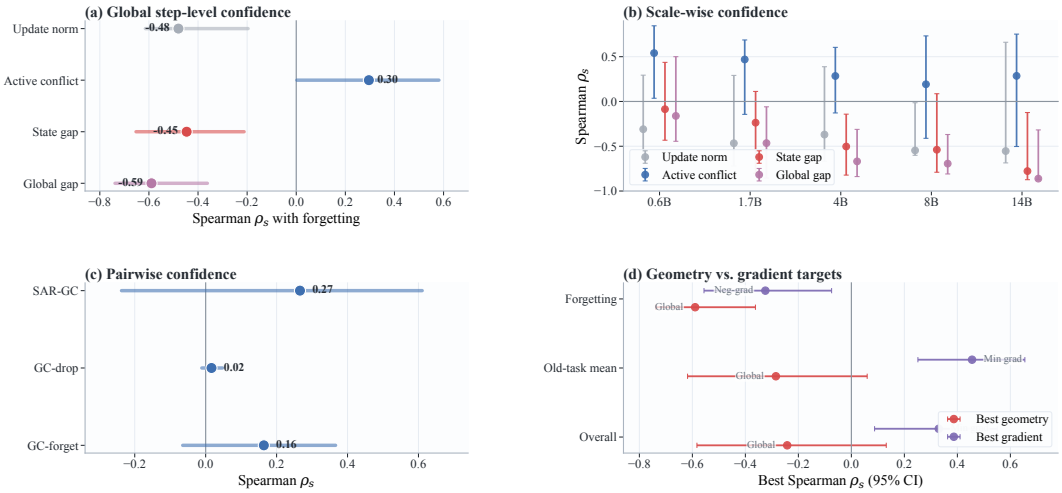


Figure 7: Statistical confidence for Sec. 3. Error bars show run-cluster bootstrap 95% confidence intervals for Spearman associations.

Table 3: Global step-level confidence for forgetting associations.

Signal	Spearman ρ_s [95% CI]	p_{perm}	n
Update norm	-0.48 [-0.61, -0.20]	<1e-3	250
Active conflict	0.30 [0.00, 0.58]	<1e-3	250
State gap	-0.45 [-0.65, -0.21]	<1e-3	250
Global gap	-0.59 [-0.74, -0.36]	<1e-3	250

Table 4: Step-level confidence by model scale.

Scale	Update norm	Active conflict	State gap	Global gap
0.6B	-0.31 [-0.64, 0.29]	0.54 [0.04, 0.84]	-0.09 [-0.43, 0.44]	-0.16 [-0.44, 0.50]
1.7B	-0.47 [-0.72, 0.29]	0.47 [-0.14, 0.69]	-0.24 [-0.80, 0.11]	-0.47 [-0.86, -0.06]
4B	-0.37 [-0.67, 0.39]	0.28 [-0.13, 0.60]	-0.50 [-0.82, -0.14]	-0.67 [-0.84, -0.31]
8B	-0.55 [-0.60, -0.01]	0.19 [-0.41, 0.73]	-0.54 [-0.79, 0.09]	-0.69 [-0.81, -0.37]
14B	-0.55 [-0.69, 0.66]	0.29 [-0.50, 0.75]	-0.78 [-0.87, -0.12]	-0.86 [-0.87, -0.32]

Table 5: Step-level confidence by continual-training method.

Method	Update norm	Active conflict	State gap	Global gap
Seq. SFT	-0.23 [-0.30, 0.18]	-0.16 [-0.49, 0.36]	-0.68 [-0.83, -0.30]	-0.70 [-0.83, -0.33]
EWC	0.35 [0.04, 0.71]	0.51 [0.08, 0.78]	-0.40 [-0.71, 0.04]	-0.42 [-0.77, 0.03]
FOREVER	-0.06 [-0.41, 0.18]	-0.27 [-0.50, 0.09]	-0.06 [-0.31, 0.34]	-0.10 [-0.33, 0.33]
AIMMerging	0.33 [0.15, 0.52]	0.22 [-0.22, 0.63]	0.10 [-0.03, 0.30]	0.01 [-0.20, 0.21]

Table 6: Global pairwise confidence for Sec. 3.2.

Pairwise relation	Spearman ρ_s [95% CI]	p_{perm}	n
SAR vs. geometry conflict	0.27 [-0.24, 0.61]	<1e-3	1820
GC vs. immediate old-task change	0.02 [-0.01, 0.05]	0.485	1820
GC vs. forgetting-from-best	0.16 [-0.06, 0.37]	<1e-3	1820

Table 7: Best geometry vs. gradient predictors by target.

Target	Best geometry predictor	Best gradient predictor	$\Delta\lvert\rho_s\rvert$ (Geom−Grad)
Forgetting	Global gap: -0.59 [-0.74, -0.36]	Neg-grad ratio: -0.32 [-0.56, -0.07]	+0.26
Old-task mean	Global gap: -0.28 [-0.62, 0.06]	Min grad-cos: 0.46 [0.25, 0.65]	-0.17
Overall	Global gap: -0.24 [-0.58, 0.13]	Min grad-cos: 0.33 [0.09, 0.55]	-0.09

pairwise level, SAR–GC is non-redundant ($\rho_s = 0.27$) but GC–drop remains near zero ($\rho_s = 0.02$, $p = 0.485$). For predictor families, geometry signals are strongest for forgetting, whereas gradient diagnostics are stronger for old-task mean and overall score, matching the complementarity claim in Sec. 3.4.

G.2 Diagnostic Dashboards for Sec. 3

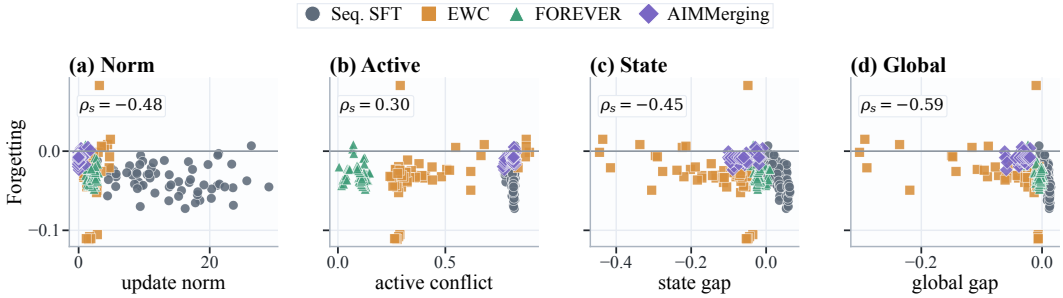


Figure 8: Drift and geometry-discrepancy signals versus forgetting. The four panels are arranged horizontally for readability. Each point is a continual post-training step; no smoothing or connecting curve is used.

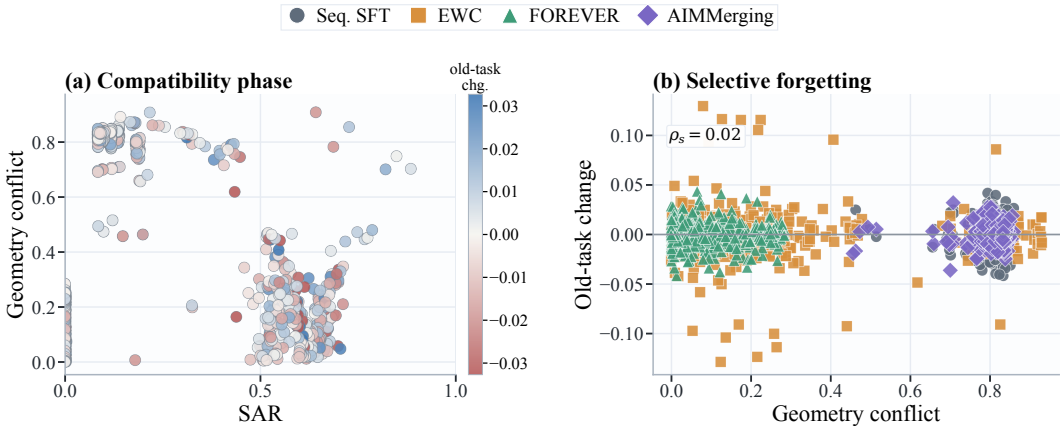


Figure 9: Pairwise subspace and geometry compatibility. The left panel compares SAR with geometry conflict, while the right panel relates pairwise geometry conflict to immediate old-task change.

Figs. 8–11 split the former Sec. 3 overview dashboard into four readable diagnostics. Fig. 8 confirms that update norm is only a coarse drift baseline: the global state-active gap has the strongest overall step-level association with forgetting ($\rho_s = -0.59$), compared with update norm ($\rho_s = -0.48$) and active-pair geometry conflict ($\rho_s = 0.30$). Fig. 9 shows that SAR and geometry conflict are related but non-redundant, while pairwise geometry conflict alone remains weak for immediate selective forgetting. Figs. 10 and 11 provide the corresponding scale/method summaries: the state-relative association strengthens on larger models, reaching -0.67, -0.69, and -0.86 for Qwen3-4B, 8B, and 14B, whereas GC-drop remains near zero across methods and scales.

Fig. 12 expands the state-relative tracking view. Under Seq. SFT, active-pair conflict often changes abruptly with local task transitions, whereas state-relative and global geometry gaps evolve with the accumulated model state and more closely track retention loss. For 1.7B/4B/8B/14B, the correlation between retention loss and state-relative conflict is $0.60/0.70/0.75/0.79$, while the corresponding global-gap association is $0.81/0.68/0.72/0.65$. The 0.6B case is noisier, consistent with the weaker and less stable small-model correlations reported in Table 8. Together, these split diagnostics provide the complete evidence behind the focused main-text figures.

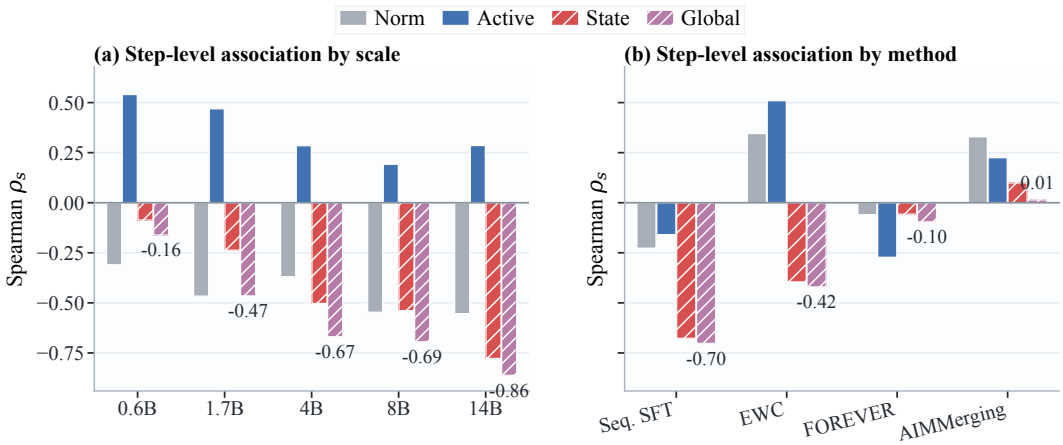


Figure 10: Step-level correlation summary by scale and method. Norm, active-pair conflict, state gap, and global gap are compared against forgetting using Spearman correlation.

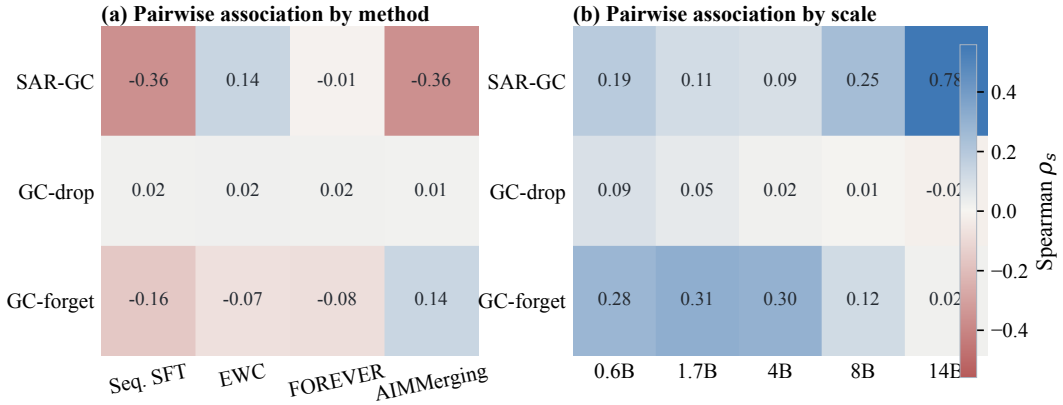


Figure 11: Pairwise correlation summary by method and scale. SAR-GC measures the relation between subspace overlap and geometry conflict; GC-drop and GC-forget measure pairwise geometry against immediate old-task change and best-previous forgetting.

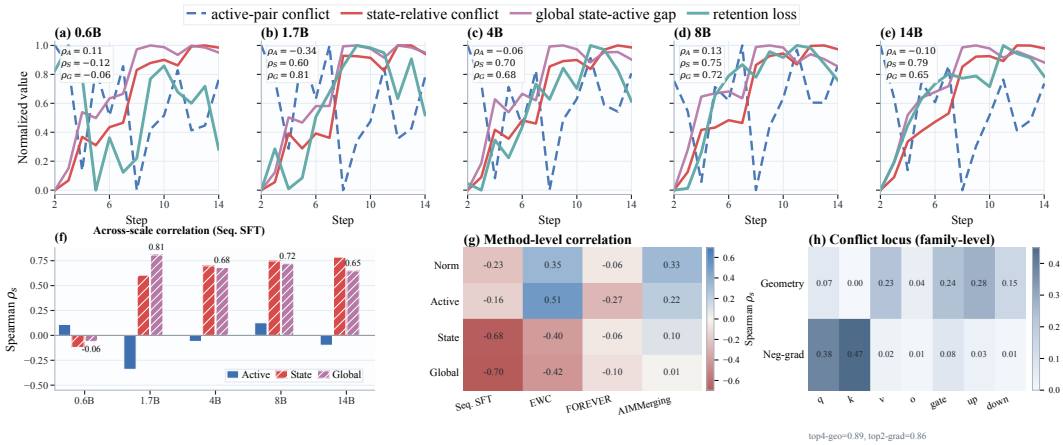


Figure 12: Full state-relative geometry diagnostic. This view expands the state-relative analysis with all Qwen3 scales, cross-scale correlations, method-level summaries, and module-level signal separation.

G.3 Additional Step-Level and State-Relative Analysis

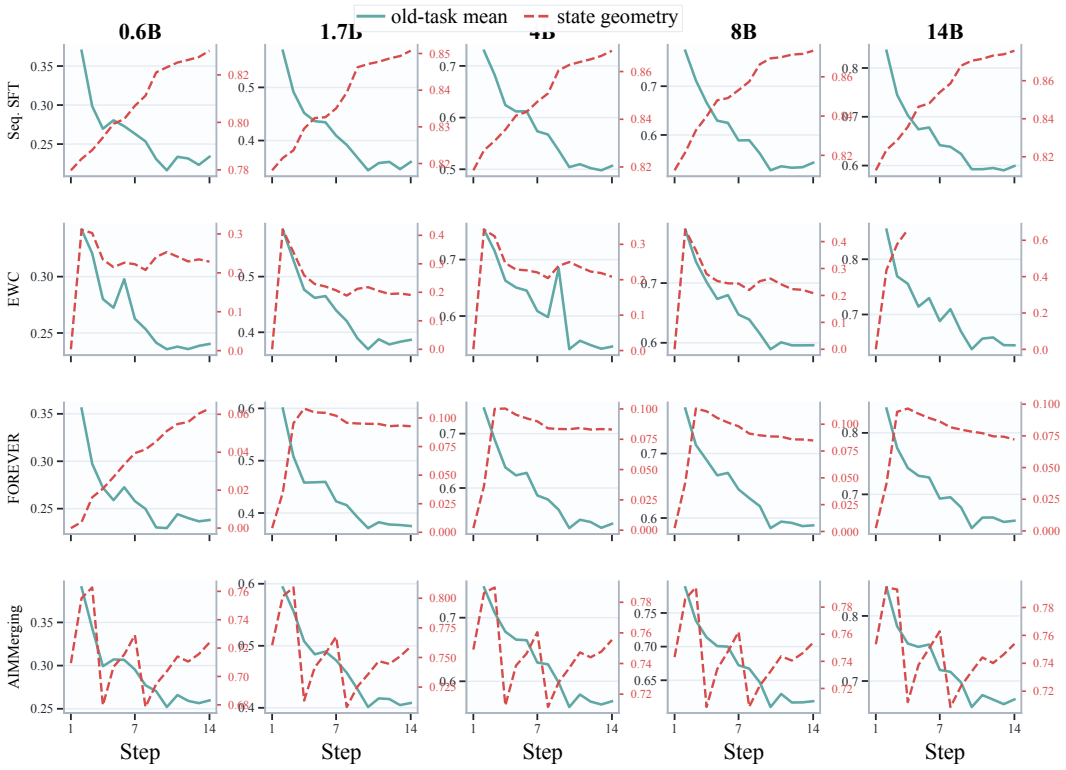


Figure 13: Step-level continual post-training dynamics by method. We show downstream retention and mechanism signals across steps for Seq. SFT, EWC, FOREVER, and AIMMerging.

Table 8: Step-level Spearman correlations with forgetting. Norm denotes incremental update norm; Active denotes active-pair geometry conflict; State and Global denote state-relative geometry gaps.

Group	Norm	Active	State	Global
All	-0.48	0.30	-0.45	-0.59
0.6B	-0.31	0.54	-0.09	-0.16
1.7B	-0.47	0.47	-0.24	-0.47
4B	-0.37	0.28	-0.50	-0.67
8B	-0.55	0.19	-0.54	-0.69
14B	-0.55	0.29	-0.78	-0.86

Table 9: Step-level Spearman correlations with forgetting by method. Norm denotes incremental update norm; Active denotes active-pair geometry conflict; State and Global denote state-relative geometry gaps.

Group	Update norm	Active GC	State gap	Global gap	Neg-grad ratio	Min grad cos
Seq. SFT	-0.23	-0.16	-0.68	-0.70	-0.05	0.13
EWC	0.35	0.51	-0.40	-0.42	-0.53	0.54
FOREVER	-0.06	-0.27	-0.06	-0.10	-0.46	0.39
AIMMerging	0.33	0.22	0.10	0.01	-0.31	0.35

Fig. 13 expands the step-level analysis by separating continual post-training methods. The main trend is that raw update magnitude captures a coarse drift effect but does not consistently explain retention across methods. For example, the global Spearman correlation between incremental update norm and forgetting is -0.48, whereas the global state-active geometry gap reaches -0.59. This gap becomes

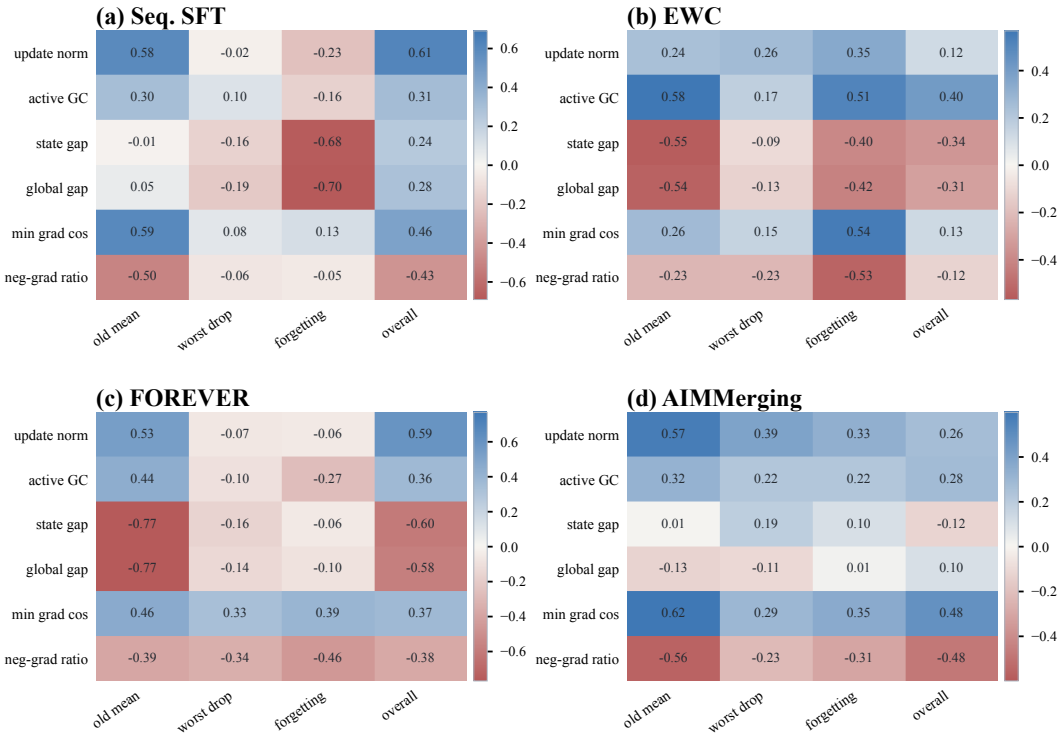


Figure 14: Method-level correlation heatmaps. We compare update norm, SAR, geometry conflict, and gradient conflict against retention, forgetting, and final performance.

Table 10: Top step-level predictors for downstream targets. We rank predictors by absolute Spearman correlation.

Target	Predictor	Pearson	Spearman	$\lvert\rho_s\rvert$
Forgetting	Merged norm	-0.41	-0.64	0.64
Forgetting	Global gap	-0.33	-0.59	0.59
Forgetting	Update norm	-0.28	-0.48	0.48
Forgetting	State gap	-0.31	-0.45	0.45
Forgetting	Global active GC	0.27	0.36	0.36
Forgetting	Neg-grad ratio	-0.16	-0.32	0.32
Forgetting	Min grad cos	0.25	0.30	0.30
Forgetting	Active GC	0.25	0.30	0.30
Old-task mean	Min grad cos	0.42	0.46	0.46
Old-task mean	Neg-grad ratio	-0.16	-0.38	0.38
Old-task mean	Global gap	-0.23	-0.28	0.28
Old-task mean	Global active GC	0.16	0.27	0.27
Old-task mean	Active SAR	0.06	-0.26	0.26
Old-task mean	Active SAR (clip)	0.06	-0.26	0.26
Old-task mean	State gap	-0.22	-0.21	0.21
Old-task mean	Active GC	0.12	0.20	0.20
Overall	Min grad cos	0.30	0.33	0.33
Overall	Neg-grad ratio	-0.18	-0.31	0.31
Overall	Active SAR	0.07	-0.26	0.26
Overall	Active SAR (clip)	0.08	-0.26	0.26
Overall	Global gap	-0.09	-0.24	0.24
Overall	State gap	-0.12	-0.23	0.23
Overall	Active GC	0.08	0.16	0.16
Overall	Global active GC	0.10	0.15	0.15

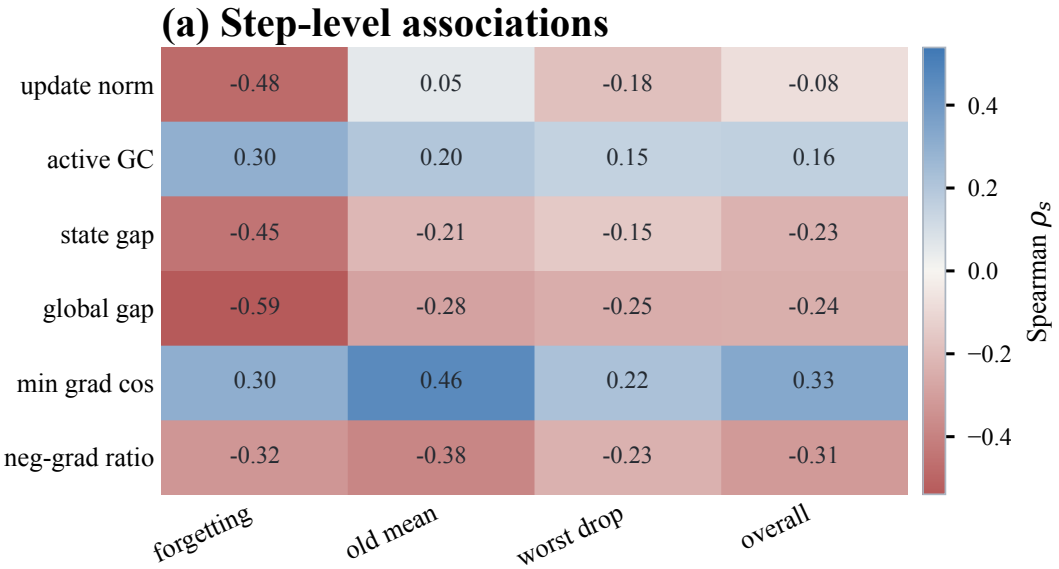


Figure 15: Global step-level explanation heatmap. Each cell reports the Spearman correlation between a mechanism signal and a downstream target.

stronger at larger scales: its correlation with forgetting is -0.67, -0.69, and -0.86 for Qwen3-4B, 8B, and 14B, respectively. By contrast, active-pair geometry conflict is more method-dependent and remains weaker as a step-level forgetting predictor.

Fig. 14 summarizes method-level correlations across mechanism signals. The heatmaps show that no single raw drift statistic consistently explains all outcomes. State-relative geometry is more aligned with forgetting in Seq. SFT and larger models, while gradient conflict contributes more to old-task retention and overall performance. This supports our use of geometry conflict as a compatibility signal, rather than treating update magnitude as a sufficient explanation.

Table 9 further decomposes the step-level correlations by method. Seq. SFT shows the clearest state-relative pattern: the state gap and global gap correlate with forgetting at $\rho_s = -0.68$ and -0.70, respectively, while update norm is weaker ($\rho_s = -0.23$). For EWC, active-pair geometry conflict and gradient conflict are more pronounced, with Active GC at $\rho_s = 0.51$ and minimum gradient cosine at $\rho_s = 0.54$. FOREVER weakens the geometry-forgetting association, consistent with replay reducing direct sequential interference. AIMMerging improves retention but also compresses the variance of forgetting, making correlations less directly interpretable.

Fig. 15 and Table 10 summarize the same trends across targets. For forgetting, the strongest geometry-based state signal is the global gap ($\rho_s = -0.59$), which is stronger than the incremental update norm ($\rho_s = -0.48$) and active-pair geometry conflict ($\rho_s = 0.30$). For old-task mean and overall performance, gradient conflict becomes more visible: minimum gradient cosine reaches $\rho_s = 0.46$ with old-task mean and $\rho_s = 0.33$ with overall score. This supports the main-text interpretation that geometry conflict is the more direct update-integration signal, while gradient conflict exposes complementary optimization-level interference.

G.4 Additional Pairwise Compatibility Analysis

Fig. 16 and Fig. 17 provide task-pair views of selective forgetting and geometry conflict. The pairwise results explain why task-task compatibility is informative but incomplete. Across 1,820 task pairs, SAR and geometry conflict have moderate global association ($\rho_s = 0.27$), but geometry conflict is only weakly correlated with immediate old-task change ($\rho_s = 0.02$). This means that pairwise conflict helps identify compatibility regimes, but forgetting is not determined by isolated task pairs alone.

Table 12 lists the largest selective-forgetting transitions. The most severe drops concentrate around adding Math under EWC for Qwen3-4B, where old-task scores drop by up to 12.9 points. These

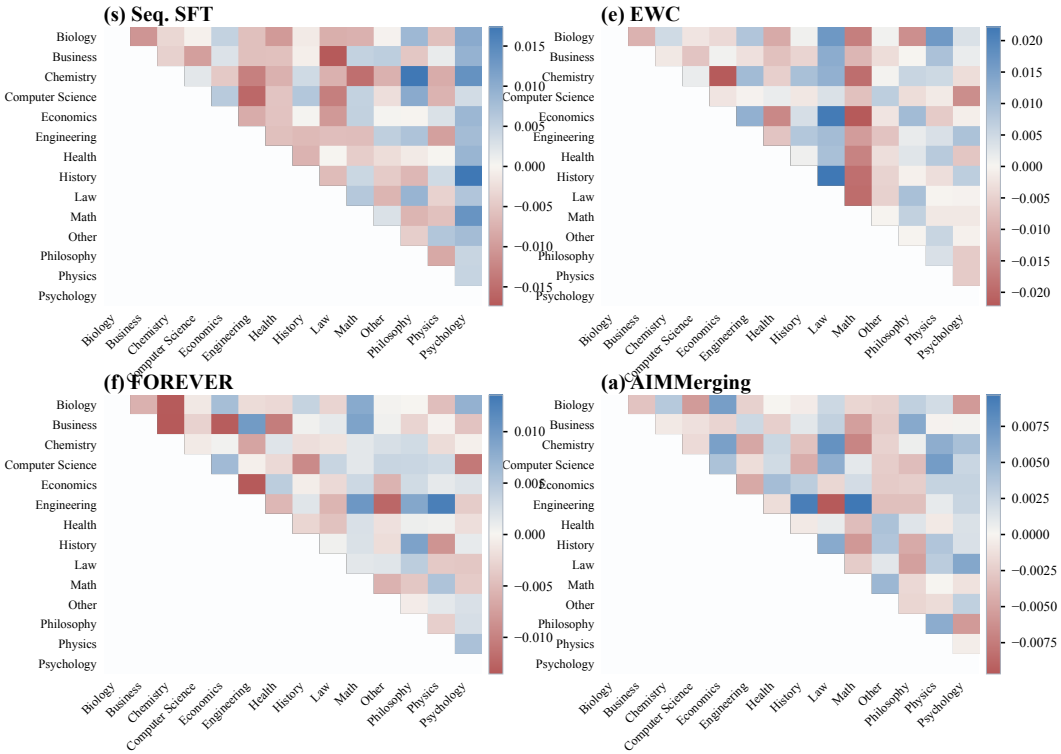


Figure 16: Task-pair selective forgetting by method. Each heatmap reports the old-task score change after introducing a new task.

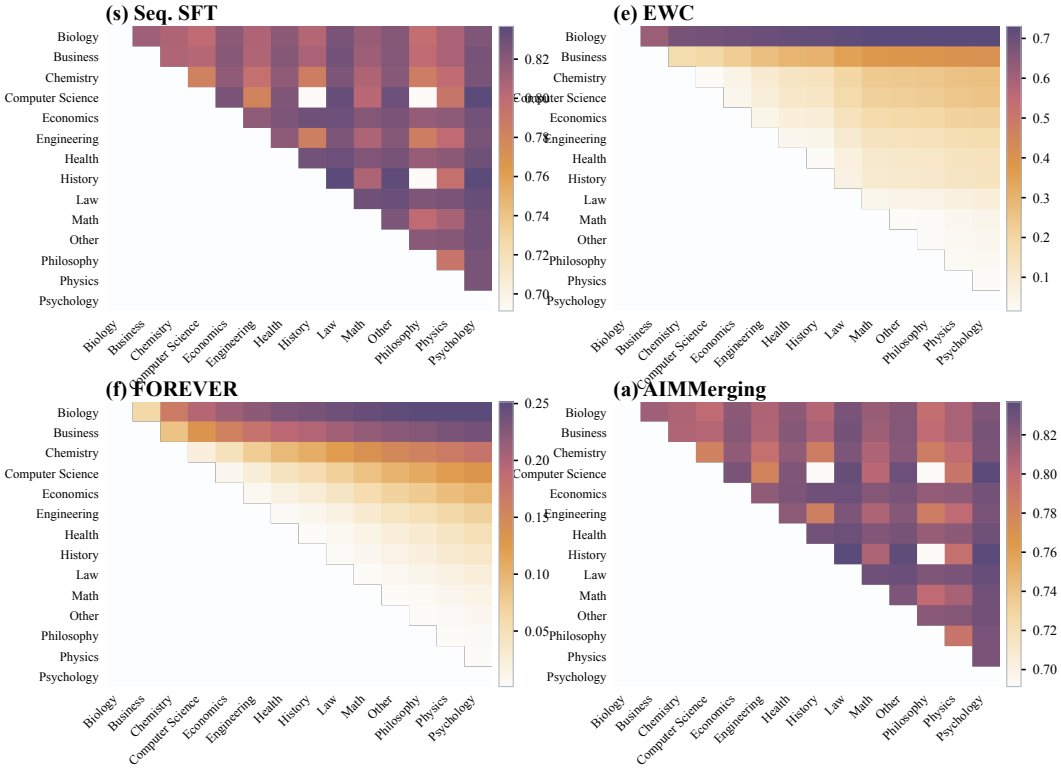


Figure 17: Task-pair geometry conflict by method. Pairwise geometry conflict reveals compatibility structure that is not captured by old-task score change alone.

Table 11: Pairwise compatibility summary by method. SAR-GC denotes Spearman correlation between SAR and geometry conflict; GC-drop denotes geometry conflict vs. immediate old-task change; GC-forget denotes geometry conflict vs. forgetting from best previous score.

Method	SAR-GC	GC-drop	GC-forget
Seq. SFT	-0.36	0.02	-0.16
EWC	0.14	0.02	-0.07
FOREVER	-0.01	0.02	-0.08
AIMMerging	-0.36	0.01	0.14

Table 12: Most harmful task transitions. Drop reports the old-task score change after adding the new task; GC is pairwise geometry conflict.

Transition	Method	Drop	GC
Math→History	EWC	-0.129	0.12
Math→Economics	EWC	-0.123	0.22
Math→Chemistry	EWC	-0.114	0.26
Math→Health	EWC	-0.104	0.14
Math→Computer Science	EWC	-0.100	0.26
Math→Law	EWC	-0.097	0.05
Math→Business	EWC	-0.093	0.44
Math→Engineering	EWC	-0.091	0.17

Table 13: Full pairwise compatibility summary by method. SAR-GC measures the relation between subspace overlap and geometry conflict; GC-drop and GC-forget measure pairwise geometry against immediate old-task change and forgetting from best previous score.

Method	N	SAR-GC	SAR-drop	GC-drop	GC-forget	Med. SAR	Med. GC
Seq. SFT	455	-0.36	0.04	0.02	-0.16	0.12	0.82
EWC	455	0.14	0.01	0.02	-0.07	0.57	0.16
FOREVER	455	-0.01	-0.04	0.02	-0.08	0.00	0.06
AIMMerging	455	-0.36	0.00	0.01	0.14	0.12	0.82

Table 14: Full pairwise compatibility summary by model scale. Pairwise compatibility is informative but does not fully explain continual forgetting.

Size	N	SAR-GC	SAR-drop	GC-drop	GC-forget	Med. SAR	Med. GC
0.6B	364	0.19	0.03	0.09	0.28	0.18	0.74
1.7B	364	0.11	0.02	0.05	0.31	0.13	0.80
4B	364	0.09	0.04	0.02	0.30	0.12	0.78
8B	364	0.25	-0.01	0.01	0.12	0.10	0.80
14B	364	0.78	-0.03	-0.02	0.02	0.04	0.37

Table 15: Global pair-level predictor ranking. Pairwise metrics are ranked by their absolute Spearman correlation with immediate old-task change and forgetting.

Target	Predictor	Pearson	Spearman	$\lvert\rho_s\rvert$
Old-task change	Min grad cos	-0.00	0.04	0.04
Old-task change	Neg-grad ratio	0.01	-0.03	0.03
Old-task change	Mean grad cos	0.02	0.03	0.03
Old-task change	GC	0.01	0.02	0.02
Old-task change	Max GC	0.01	0.01	0.01
Old-task change	Mean SAR	-0.01	0.01	0.01
Old-task change	Mean SAR (clip)	-0.01	0.01	0.01
Old-task change	Max SAR	0.00	0.00	0.00
Forgetting	Max GC	0.12	0.17	0.17
Forgetting	Mean grad cos	0.12	0.17	0.17
Forgetting	GC	0.14	0.16	0.16
Forgetting	Mean SAR (clip)	0.03	-0.16	0.16
Forgetting	Mean SAR	0.03	-0.16	0.16
Forgetting	Min grad cos	0.08	0.15	0.15
Forgetting	Max SAR	0.02	-0.14	0.14
Forgetting	Neg-grad ratio	-0.05	-0.14	0.14

Table 16: Top harmful task transitions. We report the largest old-task drops together with SAR, geometry conflict, and gradient cosine.

Size	Method	Step	Transition	Drop	Forget	SAR	GC	Grad cos
4B	EWC	10	Math→History	-0.129	-0.129	0.596	0.125	0.424
4B	EWC	10	Math→Economics	-0.123	-0.123	0.592	0.216	0.573
4B	EWC	10	Math→Chemistry	-0.114	-0.114	0.574	0.264	0.700
4B	EWC	10	Math→Health	-0.104	-0.104	0.597	0.137	0.472
4B	EWC	10	Math→Computer Science	-0.100	-0.100	0.590	0.258	0.623
4B	EWC	10	Math→Law	-0.097	-0.097	0.612	0.054	0.507
4B	EWC	10	Math→Business	-0.092	-0.092	0.550	0.440	0.706
4B	EWC	10	Math→Engineering	-0.091	-0.091	0.596	0.170	0.651
4B	EWC	10	Math→Biology	-0.091	-0.091	0.319	0.823	0.582
14B	EWC	9	Law→Computer Science	-0.058	-0.058	0.002	0.073	0.650
14B	EWC	9	Law→Chemistry	-0.050	-0.050	0.003	0.127	0.507
0.6B	EWC	2	Business→Biology	-0.049	-0.049	0.434	0.617	0.503

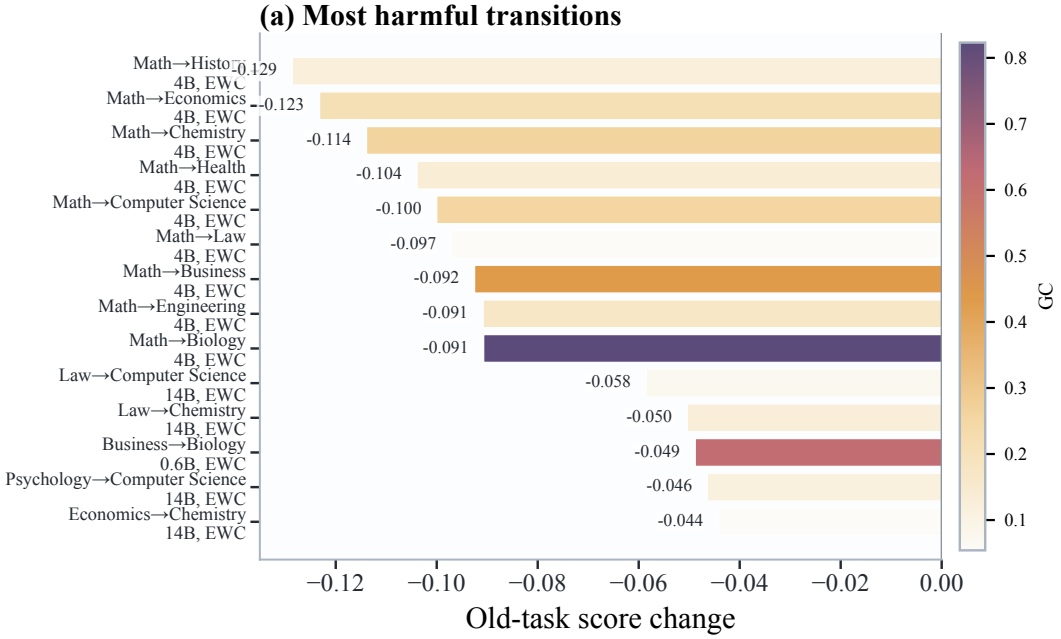


Figure 18: Most harmful task transitions. We visualize the largest old-task drops after introducing new tasks.

cases are not explained by pairwise geometry conflict alone: some harmful transitions have moderate conflict, while others have high conflict. This further motivates the state-relative analysis in Sec. 3.3.

Tables 13 and 14 expand the compact pairwise summary. Across methods, GC-drop remains weak ($\rho_s = 0.01$–0.02), confirming that immediate selective forgetting is not determined by pairwise geometry conflict alone. However, SAR and geometry conflict are not redundant: Seq. SFT and AIMMerging show negative SAR-GC correlations ($\rho_s = -0.36$), while EWC shows a weak positive relation ($\rho_s = 0.14$). Across scales, GC-forget is more visible on 0.6B, 1.7B, and 4B ($\rho_s = 0.28$, 0.31, and 0.30), but becomes weak on 8B and 14B. This scale dependence motivates the state-relative analysis in the main text.

Table 15 shows that pairwise predictors have limited explanatory power for immediate old-task change: the largest absolute Spearman correlation is only 0.04. For forgetting from best previous score, pairwise geometry conflict is more useful but still modest ($\rho_s = 0.16$ for mean GC and 0.17 for max GC). Table 16 gives concrete failure cases. The largest drops concentrate around the Math step for Qwen3-4B EWC, including Math→History (-0.129), Math→Economics (-0.123), and Math→Chemistry (-0.114). Their geometry conflict values vary widely, from 0.054 for Math→Law to 0.823 for Math→Biology, which again shows that harmful forgetting is not captured by a single pairwise metric. Fig. 18 provides the same cases as a compact visual summary.

G.5 Geometry and Gradient Conflict by Module Family

Table 17: Family-level metric summary. We report median SAR, mean geometry conflict, mean gradient cosine, and negative-gradient ratio by module family.

Family	Med. SAR	Mean GC	Mean grad cos	Neg-grad ratio
q_proj	0.20	0.56	0.59	0.03
k_proj	0.12	0.40	0.57	0.03
v_proj	0.21	0.56	0.64	0.01
o_proj	0.13	0.47	0.62	0.01
gate_proj	0.11	0.49	0.63	0.02
up_proj	0.11	0.50	0.65	0.02
down_proj	0.08	0.45	0.64	0.02

Table 18: Top-layer family distribution. We compare the module families most frequently appearing among top geometry-conflict layers and top negative-gradient layers.

Metric	q_proj	k_proj	v_proj	o_proj	gate_proj	up_proj	down_proj
Top GC layers	0.07	0.00	0.23	0.04	0.24	0.28	0.15
Top neg-grad layers	0.38	0.47	0.02	0.01	0.08	0.03	0.01

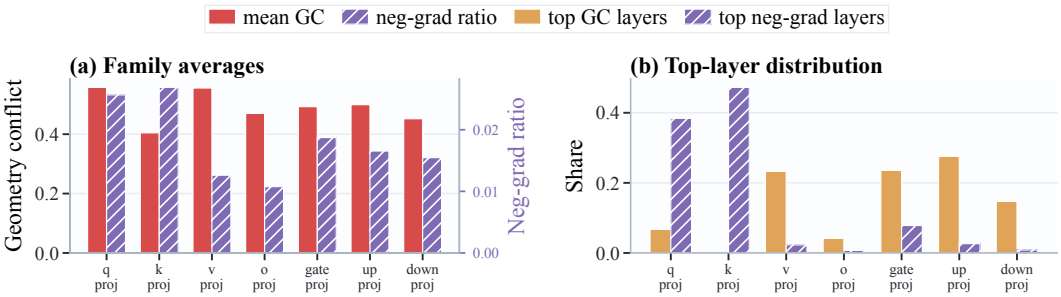


Figure 19: Family-level mechanism profile. Geometry conflict and gradient conflict emphasize different module families.

Fig. 19 and Tables 17–18 provide the module-level decomposition behind Sec. 3.4. Geometry conflict and gradient conflict are not redundant: they emphasize different module families and correspond to different failure modes.

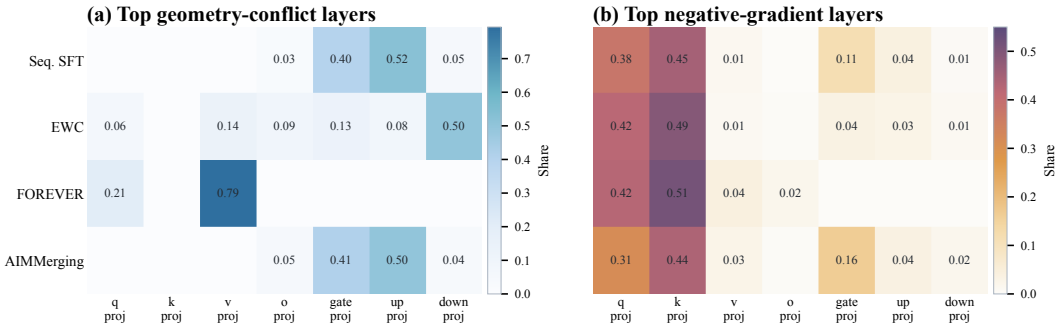


Figure 20: Method-wise top-layer family distribution. We decompose top geometry-conflict layers and top negative-gradient layers by method and module family.

At the family-average level, geometry conflict is high across both attention and MLP projections. The largest mean geometry-conflict values appear in q_proj and v_proj (0.56 for both), while MLP projections also remain high: up_proj, $gate$_proj, and $down$_proj reach 0.50, 0.49, and 0.45, respectively. This indicates that geometry conflict is not merely a sparse gradient-opposition event; it reflects broad mismatch in task-induced update structure.

The top-layer distribution shows a sharper separation. Top geometry-conflict layers concentrate in value and MLP-related modules: up_proj accounts for 0.28, $gate$_proj for 0.24, v_proj for 0.23, and $down$_proj for 0.15. Together, these four families explain 0.90 of top geometry-conflict layers. By contrast, top negative-gradient layers concentrate in attention query/key modules: k_proj accounts for 0.47 and q_proj for 0.38, together explaining 0.85 of top negative-gradient layers.

Fig. 20 further decomposes this pattern by continual post-training method. The separation between geometry conflict and gradient conflict is stable, but the geometry-conflict locus varies with the update-integration strategy. For Seq. SFT, top geometry-conflict layers concentrate almost entirely in MLP projections, especially up_proj (0.52) and $gate$_proj (0.40). AIMMerging shows a similar geometry pattern, with up_proj and $gate$_proj accounting for 0.50 and 0.41, respectively. This suggests that direct sequential update accumulation and ungated merging both induce strong mismatch in MLP transformation pathways.

EWC shifts the geometry-conflict locus toward $down$_proj, which accounts for 0.50 of top geometry-conflict layers, with smaller contributions from v_proj (0.14), $gate$_proj (0.13), and o_proj (0.09). This is consistent with regularization changing where update mismatch appears rather than removing it entirely. FOREVER shows a different pattern: top geometry-conflict layers concentrate in v_proj (0.79) and q_proj (0.21), suggesting that replay changes the geometry of retained and incoming updates, making attention-value pathways more prominent.

In contrast, top negative-gradient layers are consistently concentrated in query/key projections across all methods. For Seq. SFT, q_proj and k_proj account for 0.38 and 0.45; for EWC, 0.42 and 0.49; for FOREVER, 0.42 and 0.51; and for AIMMerging, 0.31 and 0.44. The only notable secondary contribution is $gate$_proj under AIMMerging (0.16) and Seq. SFT (0.11). Thus, gradient opposition remains largely attention-centric even when the geometry-conflict locus changes across methods.

This module-level separation supports the main-text interpretation. Geometry conflict captures mismatch in how task updates should be integrated in weight space, especially in value and MLP transformation pathways. Gradient conflict captures direct optimization opposition, most visible in attention query/key projections. The two signals therefore provide complementary views of continual post-training failure: geometry conflict is better suited as an update-integration control signal, while gradient conflict remains useful as a diagnostic for optimization-level interference.

The predictor-level results in Fig. 15 are consistent with this distinction. For forgetting, the strongest state-relative geometry signal is the global gap ($\rho_s = -0.59$), exceeding update norm ($\rho_s = -0.48$) and active-pair geometry conflict ($\rho_s = 0.30$). For retained-task and overall performance, gradient diagnostics become more visible: minimum gradient cosine reaches $\rho_s = 0.46$ with old-task mean and $\rho_s = 0.33$ with overall score, while negative-gradient ratio reaches $\rho_s = -0.38$ and -0.31, respectively. This further supports the complementary role of geometry and gradient conflict.

H Additional Experiments for Sec. 5

H.1 Experimental Setup Details

Model scales and task experts. All experiments start from the corresponding Qwen3 backbone and construct task experts relative to the same pretrained initialization. GCWM and the data-free merging baselines operate only on these task updates at integration time; they do not access replay data during merging. Seq. SFT, EWC, and FOREVER are included as reference continual-training pipelines because they respectively perform sequential optimization, regularized optimization, or replay-based training.

Training data. For the domain-continual setting, we use MMLU-Pro-CoT-Train-Labeled and sample 1k training examples for each of the 14 MMLU-Pro sub-domains. For the capability-continual setting, we construct math and code capability experts from two instruction sources: 30k examples from the math split of Nemotron-Post-Training-Dataset-v1 and 30k examples from CodeFeedback-Filtered-Instruction. This setup separates the data used to train task experts from the held-out benchmark suite used to measure capability retention and transfer.

Evaluation suite. The domain-continual setting evaluates all 14 MMLU-Pro sub-categories and reports both overall accuracy and per-domain accuracy. The capability-continual setting uses a mixed reasoning and code suite: GPQA-Diamond for graduate-level science reasoning, GSM8K for grade-school mathematical reasoning, MATH-500 for competition-style mathematics, HumanEval for Python code generation, and MMLU-Pro for broad knowledge retention. We use accuracy or exact-match style scoring for multiple-choice and math benchmarks, and pass@1 for HumanEval. The evaluation code we employ strictly adheres to the evaluation and inference configurations from the Qwen Technical Report [53], and produces results on the original Qwen3-7B and Qwen3-14B models that are aligned with the Qwen Technical Report.

Evaluation reliability. LLM benchmark scores can vary across repeated runs and execution environments [70]. Therefore, all reported capability-continual performance scores are averaged over five independent evaluation runs, using different decoding seeds where stochastic decoding is applied. We report average accuracy and keep benchmark scripts, decoding settings, and task order fixed across data-free integration methods.

Compute resources. Additional task-expert training used an internal Slurm-managed cluster with up to 64 NVIDIA H800 GPUs. The core GCWM merge, geometry construction, and scaling-law analyses are data-free and run on CPU nodes. Our CPU runs use Slurm allocations from dual-socket Intel Xeon Platinum 8480CL machines (2 sockets, 56 cores per socket, 224 logical CPUs, 210 MiB L3 cache); typical analysis jobs request a subset of cores (e.g., 24 CPU cores), while large parallel sweeps can request larger CPU allocations. GCWM introduces no additional inference-time cost after merging: the final model is evaluated with the same architecture as the corresponding merged checkpoint.

Remark: Multi-Task Learning (MTL) is treated as a joint-training upper-bound reference rather than a data-free method. In tables, bold numbers mark the best non-MTL method, while underlined MTL numbers indicate the upper-bound reference. All data-free update-integration methods are compared under the same trained-expert inputs, so differences reflect the merge-time integration rule rather than additional training data.

H.2 Evaluation Prompt

H.3 Performance Context and Data Completeness

Fig. 21 and Tables 20–21 provide performance context for the analysis. AIMMerging improves final overall performance at every scale, from 0.271 on Qwen3-0.6B to 0.678 on Qwen3-14B. It also yields consistently small final forgetting, with final forgetting scores of -0.011, -0.009, -0.006, -0.011, and -0.007 from 0.6B to 14B. These results show that update integration can mitigate forgetting, but the analysis in Sec. 3 explains why an explicit geometry-based control signal is needed: existing strategies improve retention without directly modeling state-relative geometry conflict.

Table 19: Prompt Templates for Benchmarks

Benchmark	Prompt Template
GSM8K	**System:** `You are a helpful assistant.` **User:** `Solve the following math problem step by step. The last line of your response should display the answer enclosed within ANSWER. Example: [Example-Content] question: [Question-Content] Remember to put your answer on its own line at the end in the form ANSWER (without quotes), where $ANSWER is replaced by the actual answer to the problem.`
HumanEval	**System:** `You are a helpful assistant.` **User:** `Read the following function signature and docstring, and fully implement the function described. Your response should only contain the code for this function. [Code-Content]`
Math500	**System:** `You are a helpful assistant.` **User:** `[Question-Content]`
MMLU-Pro	**System:** `You are a helpful assistant.` **User:** `Answer the following multiple choice question. The last line of your response should be of the following format: ANSWER: LETTER (without quotes) where LETTER is one of A,B,C,D,E,F,G,H,I,J. Think step by step before answering. Question: [Question-Content]`
MBPP	**System:** `You are a helpful assistant.` **User:** `[Code-Content]`
GPQA-Diamond	**System:** `You are a helpful assistant.` **User:** `Answer the following multiple choice question. The last line of your response should be of the following format: ANSWER: LETTER (without quotes) where LETTER is one of A,B,C,D. Think step by step before answering. [Question-Content]`

Table 20: Final MMLU-Pro overall score by scale and method.

Size	Seq. SFT	EWC	FOREVER	AIMMerging
0.6B	0.244	0.248	0.247	0.271
1.7B	0.370	0.400	0.385	0.418
4B	0.516	0.554	0.547	0.581
8B	0.549	0.604	0.596	0.629
14B	0.604	0.653	0.665	0.678

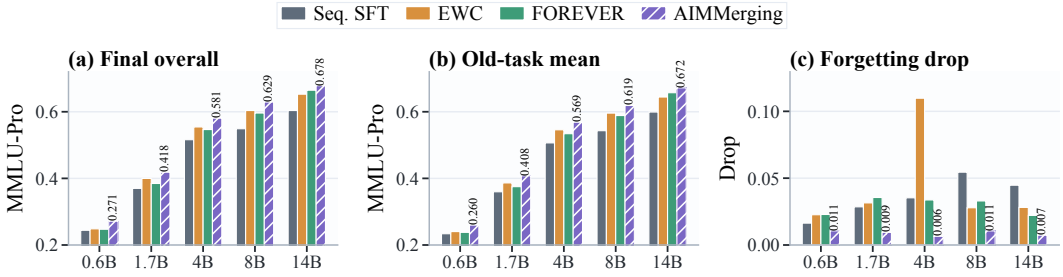


Figure 21: Final performance across model scales and continual post-training methods.

Table 21: Final performance summary. Old mean reports retained-task average at the final step; Forget reports average drop from best previous score.

Size	Method	Overall	Old mean	Forget	Current
0.6B	Seq. SFT	0.244	0.234	-0.016	0.343
0.6B	EWC	0.248	0.240	-0.023	0.284
0.6B	FOREVER	0.247	0.238	-0.023	0.296
0.6B	AIMMerging	0.271	0.260	-0.011	0.351
1.7B	Seq. SFT	0.370	0.360	-0.029	0.474
1.7B	EWC	0.400	0.386	-0.031	0.466
1.7B	FOREVER	0.385	0.375	-0.036	0.474
1.7B	AIMMerging	0.418	0.408	-0.009	0.475
4B	Seq. SFT	0.516	0.507	-0.035	0.630
4B	EWC	0.554	0.546	-0.110	0.629
4B	FOREVER	0.547	0.534	-0.034	0.635
4B	AIMMerging	0.581	0.569	-0.006	0.627
8B	Seq. SFT	0.549	0.543	-0.054	0.672
8B	EWC	0.604	0.596	-0.028	0.682
8B	FOREVER	0.596	0.589	-0.033	0.685
8B	AIMMerging	0.629	0.619	-0.011	0.679
14B	Seq. SFT	0.604	0.599	-0.045	0.726
14B	EWC	0.653	0.644	-0.028	0.717
14B	FOREVER	0.665	0.657	-0.022	0.743
14B	AIMMerging	0.678	0.672	-0.007	0.723

Table 22: Benchmark completeness and final overall scores. All benchmark files used in the analysis contain 14 evaluated continual steps after the data completion pass.

Size	Method	Steps	NaN rows	NaN values	Final overall
0.6B	Seq. SFT	14	0	0	0.244
0.6B	EWC	14	0	0	0.248
0.6B	FOREVER	14	0	0	0.247
0.6B	AIMMerging	14	0	0	0.271
1.7B	Seq. SFT	14	0	0	0.370
1.7B	EWC	14	0	0	0.400
1.7B	FOREVER	14	0	0	0.385
1.7B	AIMMerging	14	0	0	0.418
4B	Seq. SFT	14	0	0	0.516
4B	EWC	14	0	0	0.554
4B	FOREVER	14	0	0	0.547
4B	AIMMerging	14	0	0	0.581
8B	Seq. SFT	14	0	0	0.549
8B	EWC	14	0	0	0.604
8B	FOREVER	14	0	0	0.596
8B	AIMMerging	14	0	0	0.629
14B	Seq. SFT	14	0	0	0.604
14B	EWC	14	0	0	0.653
14B	FOREVER	14	0	0	0.665
14B	AIMMerging	14	0	0	0.678

Table 22 verifies that all benchmark runs used for the final analysis contain 14 steps and no missing values. This includes the re-evaluated EWC and SFT files that previously had incomplete MMLU-Pro coverage.

H.4 Comparison with Non-Continual Model Merging

We additionally compare GCWM with non-continual model merging baselines that merge task experts without explicitly modeling the sequential state. This comparison isolates a different question from the main continual setting: whether one-shot task-vector merging is sufficient when capability updates must be integrated into a shared LLM. Table 24 reports the full benchmark results, and Table 23 summarizes the overall gain over the strongest non-continual merge at each scale.

The results show two clear patterns. First, DARE is unstable in this setting: its average score collapses to 15.1% on Qwen3-0.6B, 32.2% on Qwen3-8B, and 29.6% on Qwen3-14B, with severe failures on individual benchmarks such as MBPP at 8B (0.8%) and MATH-500 at 14B (0.4%). This suggests that sparsifying or rescaling task vectors without modeling continual compatibility can be brittle for heterogeneous math, code, and knowledge updates. Second, GCWM is more effective beyond the smallest model scale. Compared with the best non-continual merge, GCWM improves average performance by $+0.21$, $+3.24$, $+5.71$, $+2.64$, and $+4.28$ points from 0.6B to 14B, respectively. The gain is also broad: GCWM matches or exceeds the best non-continual merge on 2/6, 5/6, 6/6, 4/6, and 4/6 benchmarks across the five scales. These results support the main claim that update integration benefits from an explicit geometry-aware compatibility signal rather than one-shot, state-agnostic task-vector merging.

Note: GCWM operates in parameter-update space and targets data-free continual LLM post-training. Related merging methods focus on parameter competition [71], weight evolution [72], task-vector pruning and transfer [73], modular skill transfer [74], or efficient dynamic merging [75]. Other model fusion approaches employ logit-structure distillation [68] or preference optimization [69]. In contrast, GCWM focuses on state-relative update compatibility, using geometry conflict to guide sequential update integration and mitigate forgetting in continual post-training.

Table 23: GCWM gains over the strongest non-continual merge. Wins counts the number of benchmarks where GCWM matches or exceeds the best score among TA, TIES, and DARE.

Scale	Best non-continual merge	GCWM	Gain	Wins
0.6B	TIES 39.75	39.96	+0.21	2/6
1.7B	TIES 55.02	58.26	+3.24	5/6
4B	TIES 64.56	70.27	+5.71	6/6
8B	TIES 69.90	72.54	+2.64	4/6
14B	TIES 70.04	74.32	+4.28	4/6

Table 24: Non-continual merging on capability benchmarks. Scores are accuracies or pass@1 (%). Bold marks GCWM scores that match or exceed the best non-continual merging baseline among TA, TIES, and DARE.

Method	Avg.	GPQA-D	GSM8K	HumanEval	MATH-500	MBPP	MMLU-Pro
Qwen3-0.6B							
TA	39.2	24.8	52.4	34.8	41.8	51.4	29.9
TIES	39.8	22.7	55.9	34.8	44.6	49.0	31.5
DARE	15.1	5.6	7.2	16.5	6.8	33.5	21.1
GCWM	**40.0**	23.2	51.2	**36.6**	**50.8**	50.2	27.8
Qwen3-1.7B							
TA	52.5	21.7	72.1	61.6	59.2	56.0	44.7
TIES	55.0	31.3	74.2	59.8	62.4	56.8	45.7
DARE	27.5	19.2	28.2	23.8	12.0	46.7	35.3
GCWM	**58.3**	26.3	**79.0**	**67.4**	**63.4**	**61.5**	**52.0**
Qwen3-4B							
TA	63.3	32.3	86.7	70.7	64.2	70.4	55.7
TIES	64.6	32.8	86.2	72.0	69.2	69.7	57.5
DARE	46.6	26.8	67.4	54.3	27.6	58.8	44.8
GCWM	**70.3**	**35.4**	**93.7**	**84.8**	**71.2**	**72.8**	**63.9**
Qwen3-8B							
TA	69.3	31.3	90.0	80.5	79.8	73.5	60.8
TIES	69.9	38.4	89.8	77.4	78.2	75.1	60.4
DARE	32.2	2.5	80.3	24.5	35.0	0.8	49.9
GCWM	**72.5**	**38.8**	**94.5**	**84.9**	75.4	75.0	**66.5**
Qwen3-14B							
TA	69.1	36.4	91.2	61.0	80.4	80.5	64.9
TIES	70.0	36.9	89.5	84.8	66.2	79.4	63.6
DARE	29.6	7.1	41.9	15.8	0.4	56.4	55.8
GCWM	**74.3**	**39.9**	**95.8**	**86.6**	78.2	76.7	**66.2**

H.5 Full Domain-Continual Results

We provide complete MMLU-Pro domain-continual results across all Qwen3 scales. Table 25 summarizes overall accuracy, Table 26 reports GCWM's gain over the strongest data-free update-integration baseline at each scale, and Table 27 gives the full per-domain results for Qwen3-0.6B and Qwen3-4B omitted from the main text. Together with Table 1, these tables cover all five model scales.

Across all five scales, GCWM achieves the best non-MTL overall accuracy. It improves over the strongest data-free baseline by $+0.30$, $+1.61$, $+1.19$, $+0.74$, and $+1.23$ points on Qwen3-0.6B, 1.7B, 4B, 8B, and 14B, respectively. The average overall accuracy improves from 51.1% for AIMMerging and 51.0% for OPCM to 52.3% for GCWM.

The 0.6B and 4B results further support the same trend. On Qwen3-0.6B, GCWM obtains the best non-MTL overall score (27.14%), with clear gains on chemistry, engineering, math, and other. On Qwen3-4B, GCWM improves over all data-free baselines overall (59.43%) and matches or exceeds the best non-MTL result on 12 of 14 domains, including business, chemistry, computer science, economics, health, history, law, math, other, philosophy, and physics. These results indicate that geometry-conflict-controlled integration is not only effective at larger scales, but also improves data-free update integration in small and mid-size LLM regimes.

Table 25: All-scale MMLU-Pro overall accuracy. MTL is an upper-bound reference; bold marks the best non-MTL result.

Method	0.6B	1.7B	4B	8B	14B	Avg.
MTL	27.5	44.4	61.0	65.3	68.6	53.4
Seq. SFT	24.4	36.8	51.6	55.2	60.4	45.7
EWC	24.9	40.0	55.4	60.4	65.3	49.2
FOREVER	24.7	38.5	54.7	59.6	66.5	48.8
L&S	26.2	41.1	57.3	62.4	65.6	50.5
AIMMerging	26.5	41.8	58.1	62.9	66.4	51.1
OPCM	26.8	41.7	58.2	61.9	66.6	51.0
GCWM	**27.1**	**43.5**	**59.4**	**63.7**	**67.8**	**52.3**

Table 26: GCWM gains over data-free update-integration baselines on MMLU-Pro.

Scale	Best data-free base	GCWM	Gain	Wins vs AIM
0.6B	OPCM 26.84	27.14	+0.30	7/14
1.7B	AIMMerging 41.84	43.45	+1.61	12/14
4B	OPCM 58.24	59.43	+1.19	14/14
8B	AIMMerging 62.92	63.66	+0.74	8/14
14B	OPCM 66.61	67.84	+1.23	9/14

Table 27: Additional full MMLU-Pro domain-continual results on Qwen3-0.6B and Qwen3-4B. Scores are accuracies (%). Underlined MTL is a joint-training upper-bound reference; bold marks the best non-MTL result in each block.

Method	Overall	Bio	Bus	Chem	CS	Econ	Eng	Health	Hist	Law	Math	Other	Phil	Phys	Psych
Qwen3-0.6B															
MTL	27.5	42.1	33.3	19.9	30.0	35.2	21.1	22.3	18.6	14.7	36.1	22.5	22.7	28.9	37.1
Seq. SFT	24.4	33.9	28.1	20.6	22.9	29.0	17.3	**23.5**	**18.9**	**14.4**	33.2	21.4	17.2	23.4	34.3
EWC	24.9	36.7	30.9	22.4	23.9	27.8	19.2	21.0	16.5	12.9	35.9	21.1	18.0	25.9	28.4
FOREVER	24.7	36.3	28.4	21.7	22.9	30.9	19.7	19.7	15.8	11.5	36.3	19.8	20.0	26.4	29.6
L&S	26.2	41.2	32.2	18.4	28.7	34.1	19.6	20.8	17.1	13.1	35.0	21.1	21.2	27.7	36.0
AIMMerging	26.5	42.2	32.7	18.3	29.2	34.7	19.6	20.9	17.0	12.8	35.7	21.1	21.3	28.0	36.8
OPCM	26.8	**42.3**	**33.0**	18.9	**29.5**	**35.0**	20.1	21.4	17.6	13.5	35.9	21.6	**21.8**	**28.4**	**36.9**
GCWM	**27.1**	38.9	29.1	**23.8**	28.3	34.9	**22.0**	21.1	15.2	14.3	**40.0**	**22.2**	21.2	26.4	35.1
Qwen3-4B															
MTL	61.0	81.6	67.5	67.5	65.1	70.3	52.2	57.2	50.4	35.4	71.3	51.3	53.3	65.7	69.9
Seq. SFT	51.6	73.8	57.0	49.3	51.9	61.4	44.7	50.0	41.5	24.6	62.7	43.4	44.5	53.7	63.0
EWC	55.4	**75.6**	62.9	57.9	61.5	64.5	**47.8**	53.3	42.5	25.2	69.2	46.5	45.1	57.8	62.9
FOREVER	54.7	72.1	59.4	57.4	58.8	64.5	41.2	53.3	40.4	24.7	71.0	46.7	45.7	59.3	**63.5**
L&S	57.3	72.7	64.7	62.3	62.2	66.3	46.1	56.3	41.2	26.4	75.0	46.2	45.1	61.4	61.4
AIMMerging	58.1	74.3	66.0	63.6	63.4	67.8	46.8	57.3	41.7	26.3	76.7	46.9	45.7	62.6	62.7
OPCM	58.3	75.2	65.7	63.8	63.5	67.8	47.0	57.5	42.1	27.0	76.5	47.2	46.0	62.6	63.1
GCWM	**59.4**	75.1	**66.8**	**64.4**	**64.2**	**68.6**	47.6	**58.2**	**42.6**	**27.3**	**77.5**	**47.7**	**46.6**	**63.4**	63.5

H.6 Full Capability-Continual Results

Table 28 reports full capability-continual results across all five Qwen3 scales, and Table 29 summarizes GCWM's gain over the strongest data-free baseline at each scale. Together with Table 2, these results cover both the compact main-text comparison and the full scale sweep.

Across all five scales, GCWM has the strongest average among data-free update-integration methods (63.94), ahead of OPCM (61.99), AIMMerging (60.28), and L&S (59.53). GCWM is best on overall average at 1.7B, 8B, and 14B, while remaining close to the best baseline at 0.6B and 4B. The gain over the strongest data-free baseline is -0.11, +5.78, -0.18, +1.61, and +1.39 points on 0.6B, 1.7B, 4B, 8B, and 14B, respectively.

At larger scales, GCWM shows clearer capability transfer benefits. On 14B, GCWM leads data-free baselines on GPQA-Diamond, GSM8K, HumanEval, and MMLU-Pro, while remaining competitive on MATH-500 and MBPP. At smaller scales, capability interactions are noisier: for 0.6B, GCWM is strongest on HumanEval and MBPP but slightly behind AIMMerging in average score. This pattern is consistent with the state-relative geometry findings in Sec. 3: as model capacity increases, compatibility-aware update integration yields more stable gains.

Table 28: All-scale capability-continual results. Scores are accuracies or pass@1 (%). Underlined MTL is a joint-training upper-bound reference; bold marks the best data-free update-integration method in each size block.

Method	Avg.	GPQA-D.	GSM8K	HumanEval	MATH-500	MBPP	MMLU-Pro
Qwen3-0.6B							
MTL	40.7	24.8	52.4	31.7	44.4	45.9	45.1
Seq. SFT	37.4	19.7	51.7	36.6	49.8	42.4	24.2
EWC	38.9	23.7	52.8	35.4	49.4	45.5	26.6
FOREVER	40.0	18.2	61.7	37.2	51.6	42.4	29.1
L&S	39.6	23.5	55.9	32.6	48.2	48.5	29.0
AIMMerging	**40.1**	**23.7**	**56.6**	32.9	48.8	49.0	**29.4**
OPCM	38.8	22.3	51.0	32.9	**51.2**	48.8	26.7
GCWM	40.0	23.2	51.2	**36.6**	50.8	**50.2**	27.8
Qwen3-1.7B							
MTL	57.1	26.7	76.1	61.6	63.6	56.8	57.5
Seq. SFT	51.9	18.2	70.6	64.0	64.2	59.1	35.3
EWC	54.7	24.8	75.7	61.6	67.8	57.6	40.5
FOREVER	58.3	27.3	76.7	62.2	69.6	66.9	47.3
L&S	52.4	21.3	71.0	62.8	57.5	58.4	43.5
AIMMerging	53.4	21.7	72.4	64.0	58.6	59.5	44.3
OPCM	56.8	23.2	73.0	65.2	**67.6**	59.9	51.9
GCWM	**58.3**	**26.3**	**79.0**	**67.4**	63.4	**61.5**	**52.0**
Qwen3-4B							
MTL	68.7	32.8	89.3	74.4	80.2	70.4	65.3
Seq. SFT	71.6	41.4	93.9	84.8	69.4	76.6	63.6
EWC	68.6	37.2	90.1	82.6	71.5	70.6	60.0
FOREVER	72.5	42.4	92.7	88.4	72.6	75.9	62.8
L&S	64.7	32.9	86.7	73.4	69.6	69.8	55.6
AIMMerging	65.6	33.3	87.9	74.4	70.6	70.8	56.4
OPCM	**70.5**	**37.4**	**94.5**	82.3	70.4	**73.9**	**64.2**
GCWM	70.3	35.4	93.7	**84.8**	**71.2**	72.8	63.9
Qwen3-8B							
MTL	73.3	35.9	92.2	82.9	87.2	76.3	65.4
Seq. SFT	70.3	35.9	89.3	87.2	72.6	76.6	60.4
EWC	72.6	41.3	93.5	84.7	76.5	75.4	64.2
FOREVER	75.6	47.0	94.7	92.7	69.0	82.5	68.1
L&S	69.6	32.1	88.5	79.3	76.4	75.3	66.1
AIMMerging	70.2	32.3	89.2	79.9	**77.0**	**75.9**	**66.7**
OPCM	70.9	37.8	**94.6**	79.3	74.4	74.3	65.2
GCWM	**72.5**	**38.8**	94.5	**84.9**	75.4	75.0	66.5
Qwen3-14B							
MTL	74.6	33.3	92.1	84.2	87.8	80.5	69.8
Seq. SFT	70.4	43.4	95.8	86.6	66.2	63.4	67.2
EWC	73.5	43.4	95.4	86.0	68.0	78.6	69.4
FOREVER	75.9	54.0	96.4	87.2	69.2	75.9	72.8
L&S	71.3	38.4	94.1	83.7	76.5	78.5	56.8
AIMMerging	72.2	38.8	95.2	84.7	77.4	**79.4**	57.5
OPCM	72.9	38.0	94.5	80.5	**79.7**	78.6	66.3
GCWM	**74.3**	**39.9**	**95.8**	**86.6**	78.2	76.7	**68.8**

Table 29: GCWM gains over the strongest data-free capability baseline at each scale.

Scale	Best data-free base	GCWM-Diamond	Gain	Wins vs AIM
0.6B	AIMMerging 40.07	39.96	-0.11	3/6
1.7B	OPCM 56.82	58.26	+1.44	6/6
4B	OPCM 70.45	70.27	-0.18	6/6
8B	OPCM 70.93	72.54	+1.61	3/6
14B	OPCM 72.93	74.32	+1.39	5/6

I Additional Ablation Study

Domain-continual ablation. All variants use the same Qwen3-0.6B domain-continual task experts and evaluation protocol, differing only in the integration rule.

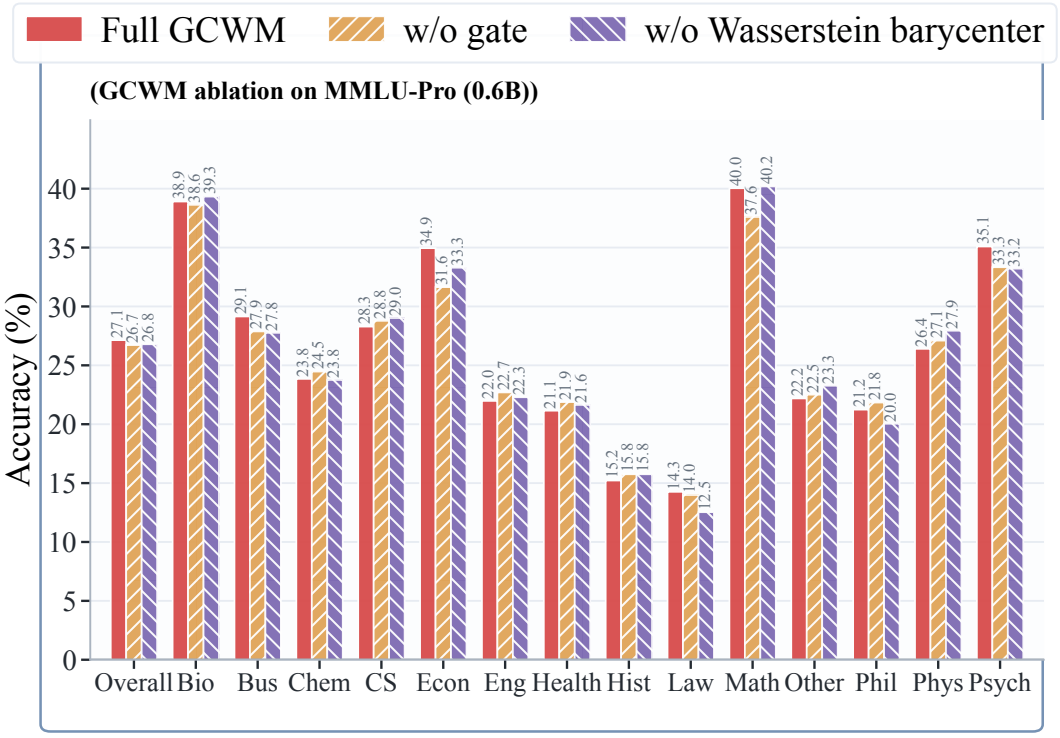


Figure 22: GCWM ablation on Qwen3-0.6B MMLU-Pro.

Fig. 22 shows that full GCWM achieves the best aggregate score, improving overall accuracy from $26.7/26.8\%$ to 27.1%. The domain breakdown shows a trade-off rather than uniform dominance: removing the gate notably hurts economics, math, and psychology, whereas replacing the Wasserstein metric weakens business, law, and psychology.

Capability-continual ablation. We provide additional capability-continual ablations on Qwen3-1.7B and Qwen3-8B. The variants isolate the two merge-time components used by GCWM: the geometry-conflict gate and the Wasserstein shared metric. The *w/o gate* variant removes conflict-conditioned interpolation and applies the geometry-aware branch uniformly, while *w/o WB* replaces the Wasserstein barycenter with a simpler shared metric. All variants use the same task experts and evaluation protocol; only the merge rule changes.

Fig. 5 visualizes the benchmark-level ablations, and Table 30 summarizes the aggregate effect. On Qwen3-1.7B, full GCWM improves the average score from 56.68% without the gate and 56.95% without the Wasserstein barycenter to 58.26%. On Qwen3-8B, the gap is larger: full GCWM reaches 72.54%, compared with 67.93% without the gate and 68.81% without the Wasserstein barycenter. This indicates that both components matter more strongly in the larger-scale capability setting, where math, code, and knowledge updates interact more sharply.

Table 31 gives the full benchmark breakdown. The improvements are not uniform across all benchmarks, which is expected for heterogeneous capability updates. For Qwen3-1.7B, removing the gate can improve MATH-500 and removing the Wasserstein barycenter can improve HumanEval, but both ablations substantially reduce MMLU-Pro and GPQA-Diamond, leading to weaker aggregate performance. For Qwen3-8B, the gate is especially important for MATH-500 ($+14.0$ points over w/o gate), while the Wasserstein barycenter is important for GSM8K, HumanEval, MMLU-Pro, and GPQA-Diamond. Overall, these ablations support the design of GCWM as a compatibility-controlled

merge: the gate decides how strongly to trust geometry-aware integration, and the Wasserstein shared metric provides the common space in which heterogeneous task updates can be combined.

Table 30: Capability ablation summary. Drop is the average-score difference between full GCWM and the corresponding ablated variant.

Scale	Full GCWM	w/o gate	Drop	w/o WB	Drop
1.7B	58.26	56.68	+1.58	56.95	+1.31
8B	72.54	67.93	+4.61	68.81	+3.73

Table 31: Capability ablation breakdown on Qwen3-1.7B and 8B. Scores are accuracies or pass@1 (%). Bold marks the best variant within each scale and metric.

Variant	Avg.	GPQA-D	GSM8K	HumanEval	MATH-500	MBPP	MMLU-Pro
Qwen3-1.7B							
GCWM	**58.3**	**26.3**	**79.0**	67.4	63.4	**61.5**	**52.0**
w/o gate	56.7	22.7	76.3	70.7	**67.0**	59.9	43.4
w/o WB	57.0	21.2	77.2	**76.8**	64.2	59.5	42.7
Qwen3-8B							
GCWM	**72.5**	**38.8**	94.5	**84.9**	75.4	75.0	**66.5**
w/o gate	67.9	34.8	**95.0**	81.7	61.4	71.2	63.4
w/o WB	68.8	35.4	88.9	76.8	**76.8**	**76.6**	58.3

J Runtime and Memory Profiling

We profile GCWM merge-time cost on Qwen3-8B and Qwen3-14B under the capability-continual setting. The purpose is to measure practical feasibility rather than training efficiency: GCWM is a data-free merge-time method, so its overhead is incurred once during model integration and adds no inference-time cost. Profiling is performed on a Slurm job using one visible GPU on node `kb3-a1-nv-dgx02`; each continual step is timed separately and peak GPU memory is recorded. The active update count m increases with the number of merged updates, while the retained rank is fixed at $r = 16$. The 8B profile covers the full three-step sequence (MMLU, math, code), including the $m = 3$ code step. For 14B, we report the two profiled steps (MMLU and math), which provide a matched $m = 1, 2$ comparison against 8B and isolate the scale effect without treating 14B as a full three-step average.

Fig. 23 visualizes the merge-time decomposition and peak memory trend, and Tables 32–33 report the exact summary and per-step values. On Qwen3-8B, the average merge step takes 40.5 ± 19.7 minutes with 7.8 ± 3.4 GB peak GPU memory across the three profiled steps. On Qwen3-14B, the two matched profiled steps take 76.2 ± 34.8 minutes with 11.7 ± 4.7 GB peak GPU memory. The cost grows with active updates because the union rank increases with m, but the expensive SPD operations remain in the projected rank-r or union-rank space rather than in the full input dimension. In practice, the dominant wall-clock terms are SVD/metric preparation and inner merge optimization; Bures–Wasserstein barycenter and matrix square-root operations are comparatively small in the profiled runs. For example, averaged over the available steps, barycenter plus matrix square-root/inverse-square-root takes 52.1 seconds on 8B and 30.5 seconds on 14B.

The implementation follows the projected low-rank formulation in Sec. 4. For an update matrix with input dimension d_{in}, output dimension d_{out}, active updates m, retained rank r, and barycenter iterations I, the dominant per-layer cost can be summarized as

$$O(md_{\text{out}}d_{\text{in}}r) + O(Ir^3) + O(md_{\text{out}}r^2),$$

up to implementation constants and merge-optimizer iterations. This highlights why GCWM is feasible at 8B/14B scales: the Bures–Wasserstein and square-root operations are applied to low-rank projected SPD matrices rather than dense $d_{\text{in}} \times d_{\text{in}}$ operators.

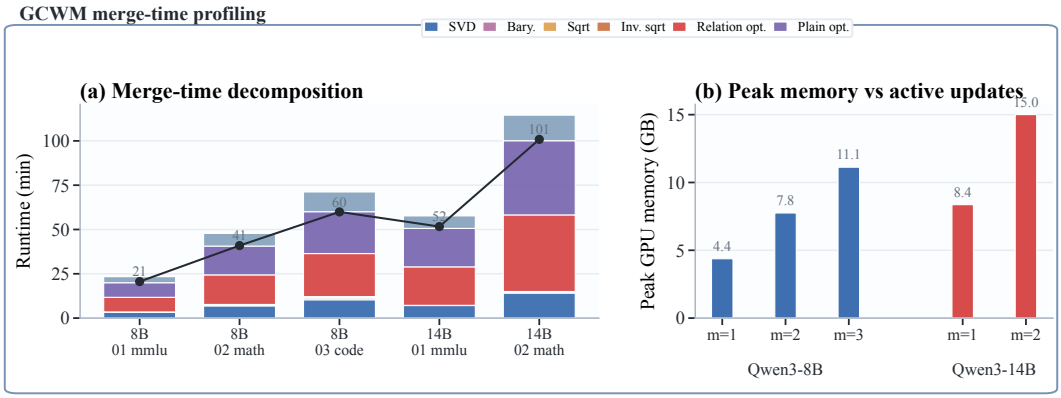


Figure 23: GCWM merge-time runtime and memory profiling. Runtime is decomposed by major merge-time components, and memory is reported as peak allocated GPU memory.

Table 32: GCWM runtime and memory summary. Values are mean ± standard deviation across the profiled continual steps; 14B uses the two matched $m = 1, 2$ steps.

Model	Steps	Layers	m	r	Union rank	Wall time (min/step)	Peak mem. (GB)	Bary.+sqrt (s)
8B	3	252	2.0	16	32	40.5 ± 19.7	7.8 ± 3.4	52.1
14B	2	280	1.5	16	24	76.2 ± 34.8	11.7 ± 4.7	30.5

Table 33: Per-step GCWM profiling. m denotes the number of active updates in the merge step.

Model	Step	m	Union rank	Wall time (min)	Peak mem. (GB)	Bary. (s)
8B	01_mmlu	1	16	20.6	4.4	0.6
8B	02_math	2	32	40.9	7.8	25.2
8B	03_code	3	48	59.9	11.1	55.0
14B	01_mmlu	1	16	51.6	8.4	3.9
14B	02_math	2	32	100.8	15.0	28.4

K Hyperparameter Sensitivity

We evaluate one-dimensional hyperparameter sweeps on Qwen3-8B under the capability-continual setting. Each sweep changes one parameter while keeping the remaining GCWM configuration fixed, which is intended to test robustness rather than tune on the evaluation set. We vary the retained geometry energy, the conflict threshold τ, the retained SVD rank r, the gate sharpness κ, and the outer merge coefficient η_t. The default configuration corresponds to energy 0.90, $\tau = 0.12$, $r = 16$, $\kappa = 10$, and $\eta_t = 0.1$.

Fig. 24 summarizes the sensitivity profile, while Tables 34–35 provide exact values. GCWM is stable under moderate choices of the geometry and gate parameters. Changing τ over $\{0.08, 0.12, 0.18\}$, r over $\{8, 16, 32, 64\}$, and κ over $\{5, 10, 20\}$ changes the average score by only 1.6, 0.9, and 1.4 points, respectively. The energy threshold has a larger but interpretable trade-off: energy 0.95 gives the best average score (72.2%), while the default energy 0.90 is more conservative on MMLU-Pro and GPQA-Diamond. The dominant sensitivity is the outer merge coefficient η_t: moderate values remain competitive, with $\eta_t = 0.3$ achieving 70.8%, but overly aggressive integration collapses performance at $\eta_t = 1.0$ (34.3%). This supports the use of a conservative validation-free coefficient schedule for data-free continual update integration.

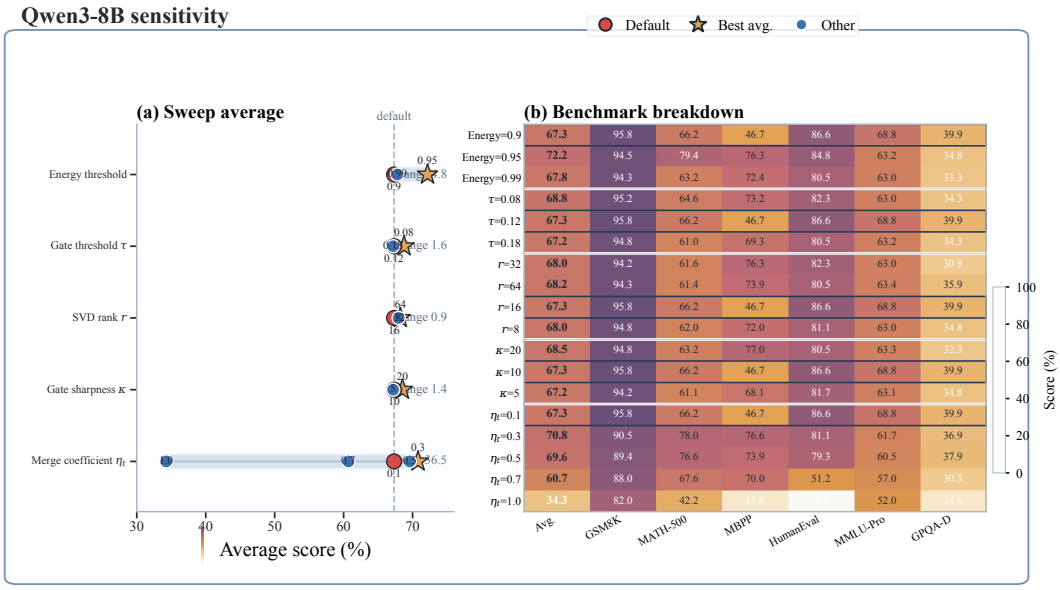


Figure 24: GCWM hyperparameter sensitivity on Qwen3-8B. Each sweep changes one parameter while keeping the others fixed; default rows are outlined and best average settings are marked by stars.

Table 34: GCWM hyperparameter sensitivity summary. Scores are average capability-continual performance (%); range is best minus worst within each one-dimensional sweep.

Sweep	Default	Best avg.	Worst avg.	Range
Energy threshold	0.9 (67.3)	**0.95 (72.2)**	0.9 (67.3)	4.8
Gate threshold τ	0.12 (67.3)	**0.08 (68.8)**	0.18 (67.2)	1.6
SVD rank r	16 (67.3)	**64 (68.2)**	16 (67.3)	0.9
Gate sharpness κ	10 (67.3)	**20 (68.5)**	5 (67.2)	1.4
Merge coefficient η_t	0.1 (67.3)	**0.3 (70.8)**	1.0 (34.3)	36.5

Table 35: Full Qwen3-8B hyperparameter sensitivity breakdown. Scores are accuracies or pass@1 (%); shaded rows denote the default setting for each sweep.

Sweep	Setting	Avg.	GSM8K	MATH-500	MBPP	HumanEval	MMLU-Pro	GPQA-D
Energy threshold	0.9	67.3	95.8	66.2	46.7	86.6	68.8	39.9
	0.95	**72.2**	94.5	79.4	76.3	84.8	63.2	34.8
	0.99	67.8	94.3	63.2	72.4	80.5	63.0	33.3
Gate threshold τ	0.08	**68.8**	95.2	64.6	73.2	82.3	63.0	34.3
	0.12	67.3	95.8	66.2	46.7	86.6	68.8	39.9
	0.18	67.2	94.8	61.0	69.3	80.5	63.2	34.3
SVD rank r	32	68.0	94.2	61.6	76.3	82.3	63.0	30.8
	64	**68.2**	94.3	61.4	73.9	80.5	63.4	35.9
	16	67.3	95.8	66.2	46.7	86.6	68.8	39.9
	8	68.0	94.8	62.0	72.0	81.1	63.0	34.8
Gate sharpness κ	20	**68.5**	94.8	63.2	77.0	80.5	63.3	32.3
	10	67.3	95.8	66.2	46.7	86.6	68.8	39.9
	5	67.2	94.2	61.1	68.1	81.7	63.1	34.8
Merge coefficient η_t	0.1	67.3	95.8	66.2	46.7	86.6	68.8	39.9
	0.3	**70.8**	90.5	78.0	76.6	81.1	61.7	36.9
	0.5	69.6	89.4	76.6	73.9	79.3	60.5	37.9
	0.7	60.7	88.0	67.6	70.0	51.2	57.0	30.3
	1.0	34.3	82.0	42.2	13.6	1.2	52.0	14.6

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's core claims (state-relative geometry explanation and GCWM as a data-free integration method), and these are supported by the empirical/theoretical sections (Secs. 3–5 and Appendix).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Appendix A, including benchmark scope, the interpretation of geometry conflict, replay-vs-data-free tradeoffs, and offline compute overhead.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Theoretical assumptions, statements, and proofs are provided in the main text and Appendix (Sec. 4 and Appendices A–B), including theorem assumptions and full derivations.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The draft specifies the continual setup, metrics, baselines, and evaluation protocol, and provides implementation/analysis details in Appendix (Secs. 5.1 and Appendix H.1).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Appendix B provides an anonymized code and data availability link and states that scripts, configurations, processed tables, and task-split reconstruction instructions are included.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training/evaluation setup, model scales, task construction, benchmarks, and baseline definitions are documented in Sec. 5.1 and expanded in Appendix H.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report statistical confidence for key Sec. 3 claims via run-cluster bootstrap confidence intervals and permutation tests (Appendix G.1, Fig. 7, Tables 3–7).

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix H.1 reports the GPU resources used for expert training and the CPU resources used for data-free merging, geometry construction.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research is conducted on open benchmarks and model post-training settings and follows NeurIPS ethical expectations. No human-subject data collection or deceptive deployment study is involved.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: Appendix B includes a concise broader-impact discussion covering data-free continual adaptation, possible misuse of easier post-training, and the need for downstream safety checks.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This submission does not release a new high-risk foundation model or scraped dataset artifact; it studies update-integration behavior on existing model families and benchmarks.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper credits upstream datasets/models/benchmarks in Sec. 5.1 and citations. We use publicly available assets with standard research usage terms and will add explicit license names/URLs in Appendix for clarity.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce a new standalone dataset or model asset release in the current submission package.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The work does not involve crowdsourcing tasks or human-subject experiments.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The work does not involve human-subject research requiring IRB review.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are used for writing, editing, and formatting.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.