MOSAIC: CONTINUAL LORA MERGING VIA HYPERNETWORK STRUCTURE

Anonymous authors
Paper under double-blind review

ABSTRACT

Hypernetworks offer an efficient alternative to LoRA fine-tuning by directly generating adapters from conditioning inputs, such as document chunks or task descriptions. However, in streaming settings, retaining every generated LoRAs increases storage and leaves information distributed across separate LoRAs. Existing weight-space merging approaches consolidate these adapters without exploiting their shared hypernetwork, whose decoder relates generating coordinates to adapter behavior. To address this, we propose MOSAIC, a continual merging method that uses the hypernetwork structure to merge generated LoRAs in hypernetwork coordinates. MOSAIC measures adapter differences through their layer-output contributions on frozen target-model activation probes and pulls the resulting activation-induced metric back to hypernetwork coordinates through the frozen decoder's Jacobian, yielding a matrix-weighted barycenter formulation with additive sufficient statistics. This formulation enables one-pass updates and order-invariant accumulation with memory independent of stream length, producing a single deployable LoRA after each arrival. MOSAIC further applies covariance shrinkage to stabilize the accumulated functional metric. Experiments across multiple hypernetwork-based LoRA generation families and target LLMs demonstrate the effectiveness of MOSAIC in continually merging generated LoRAs.

1 INTRODUCTION

Hypernetworks (Ha et al., 2017) enable rapid LLM adaptation by generating LoRA (Hu et al., 2022) in a forward pass, without per-input fine-tuning (Charakorn et al., 2025; 2026; Yu et al., 2026). Text-to-LoRA generates LoRAs from natural-language task descriptions (Charakorn et al., 2025), and Doc-to-LoRA internalizes documents into adapters (Charakorn et al., 2026). However, document chunks and new tasks often arrive sequentially, producing separate adapters whose information may be needed together at inference. For example, answering a multi-hop question (Yang et al., 2018) may depend on facts spread across several chunks, making one chunk's adapter insufficient. Similarly, when user queries lack explicit task labels, serving them with separate task adapters requires routing suitable adapters (Ostapenko et al., 2024). These challenges motivate continually consolidating generated LoRAs into a single adapter that serves the accumulated documents or tasks without revisiting earlier inputs or adapters.

Model merging offers a natural approach to this consolidation. Weight-space merging methods combine parameter updates (Yadav et al., 2023; Ilharco et al., 2023), activation-aware methods match layer outputs on probe activations (Jin et al., 2023), and LoRA-specific methods exploit the structure of low-rank updates (Stoica et al., 2025). Many methods assume simultaneous access to the models, while continual merging methods incorporate models as they arrive (Tang et al., 2025; Yang et al., 2026; Lin & Kang, 2026). However, these approaches do not explicitly exploit the shared generator of hypernetwork-generated LoRAs. Each generated adapter is decoded from a coordinate representation through a shared, frozen decoder. This structure offers two advantages. First, merging in coordinate space keeps the resulting adapter within the decoder's output family. Second, the decoder's Jacobian connects coordinates to adapter behavior, enabling functional statistics to be accumulated in coordinate space. This motivates our central question:

How can we exploit a frozen hypernetwork to continually merge generated LoRAs into a single fixed-rank adapter?

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

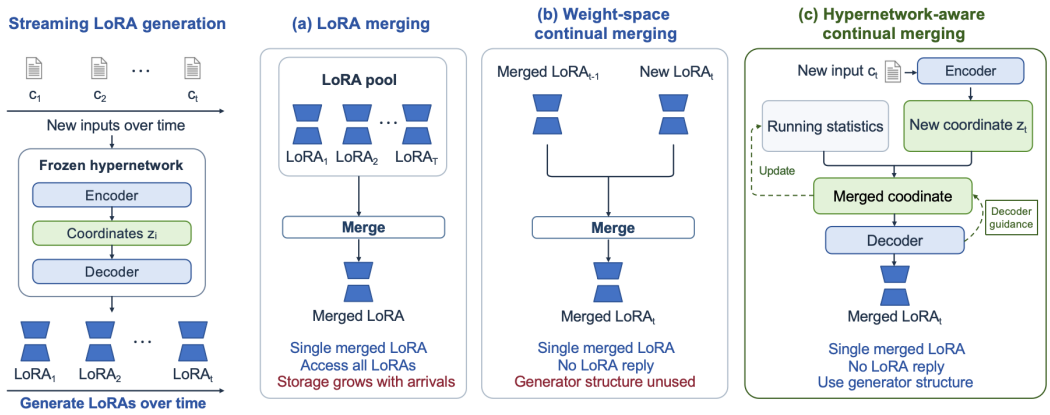


Figure 1: Continual merging of hypernetwork-generated LoRAs. A frozen hypernetwork encodes each incoming input into coordinates and decodes them into a LoRA. (a) LoRA merging retains all incoming LoRAs before consolidation. (b) Weight-space continual merging updates a single LoRA without using the generator structure. (c) Hypernetwork-aware continual merging uses the decoder structure to guide merging in coordinate space, maintains running statistics, and decodes the merged coordinates into a single LoRA.

To address the above challenges, we propose **MOSAIC** (**M**erging **O**f **S**treaming Lo**RA** **I**n Hypernetwork **C**oordinates), a continual merging framework for hypernetwork-generated LoRAs. Unlike Doc-to-LoRA's rank concatenation (Charakorn et al., 2026), which uses multi-chunk hypernetwork training and grows the adapter rank with each chunk, MOSAIC consolidates adapters from a frozen hypernetwork at a fixed output rank without additional training. MOSAIC measures adapter discrepancies through layer-output changes on frozen target-model activation probes and pulls this metric back to hypernetwork coordinates through the decoder Jacobian. Under a local linearization, the objective defines a matrix-weighted barycenter, where coordinate directions are weighted by their estimated functional importance. MOSAIC further applies Oracle Approximating Shrinkage (OAS) (Chen et al., 2010) to improve numerical conditioning. We evaluate MOSAIC across document- and task-conditioned generation on four target LLMs. In document-conditioned merging, MOSAIC achieves the best LongBench (Bai et al., 2024) average at all three evaluated chunk sizes on Qwen3-4B (Yang et al., 2025) and remains competitive on Gemma-2-2B (Team et al., 2024) and Mistral-7B (Jiang et al., 2023). In task-conditioned merging, it surpasses individual adapters in final average performance and achieves positive mean backward transfer on both Mistral-7B and Qwen3-8B (Yang et al., 2025). We also measure memory and latency costs to characterize the trade-off between performance and overhead.

Summary of contributions. This paper makes the following key contributions:

- We formulate continual merging of hypernetwork-generated LoRA in their generator coordinates, using a decoder pullback of an activation-induced functional metric.
- We derive a matrix-weighted quadratic barycenter with additive sufficient statistics, and incorporate covariance shrinkage for numerical stability.
- We evaluate MOSAIC across document- and task-conditioned LoRA generation on four target LLMs, demonstrating performance gains over continual merging baselines and examining the contributions of its components through ablation studies.

2 Related Works

2.1 Hypernetwork-Generated Adaptation

Hypernetworks provide a framework for learning how to adapt models by generating adaptation parameters from conditioning information (Ha et al., 2017). For LLMs, this approach enables LoRA generation from inputs, reducing the need for separate gradient-based fine-tuning for each new in-

put. Early work explores hypernetwork-based adaptation using learned task-specific representations to generate adapter parameters (Mahabadi et al., 2021) or prompts (He et al., 2022), while Hyperdecoders (Ivison & Peters, 2022) and HyperTuning (Phang et al., 2023) investigate adaptation conditioned on input sequences and task examples. Recent work extends to LoRA generation from diverse textual inputs: Text-to-LoRA maps natural-language task descriptions to LoRA parameters (Charakorn et al., 2025), Doc-to-LoRA generates LoRA adapters that internalize document contexts (Charakorn et al., 2026), and LatentSkill studies hypernetwork-generated LoRAs for textual agent skills (Yu et al., 2026). These approaches motivate the setting studied by MOSAIC, which addresses the complementary problem of continually consolidating adapters generated by a frozen hypernetwork as new conditioning inputs arrive.

2.2 Model Merging, LoRA Merging, and Continual Merging

Model merging consolidates adapted models into a single model without ensemble inference overhead. Model Soups averages fine-tuned checkpoints obtained from a shared initialization (Wortsman et al., 2022), while Task Arithmetic combines task vectors defined as differences between fine-tuned and pretrained weights (Ilharco et al., 2023). To reduce interference between updates, TIES-Merging removes small parameter changes and resolves sign conflicts (Yadav et al., 2023). RegMean incorporates functional information, using activation statistics to minimize discrepancies between linear-layer outputs of merged and individual models (Jin et al., 2023). For LoRA adapters, merging methods additionally exploit structure of low-rank updates. KnOTS uses joint singular value decomposition to align LoRA updates before applying merging methods (Stoica et al., 2025). LoRA-LEGO groups and clusters rank-wise parameter units across adapters, then assembles a merged LoRA from cluster centroids (Zhao et al., 2025).

Continual merging incorporates task-specific models sequentially while preserving earlier capabilities. OPCM uses orthogonal projections and adaptive scaling to integrate successive models with memory independent of stream length (Tang et al., 2025). DOP combines data-free dual orthogonal projections with multi-objective optimization to balance stability and plasticity (Yang et al., 2026). ODE-M formulates each merge as a parameter-space trajectory, using calibration feedback and a utility-aware time (Lin & Kang, 2026). Our proposed MOSAIC addresses the complementary setting in which incoming LoRAs are generated by a frozen hypernetwork, expressing functional discrepancies in generator's coordinates and accumulating additive statistics that support continual consolidation without retaining previous inputs or adapters.

3 Preliminaries and Problem Setup

In this section, we introduce hypernetwork-generated LoRA and then formulate the continual merging problem addressed by MOSAIC.

Hypernetwork-generated LoRA. For each adapted layer $l \in \mathcal{L}$, let $\boldsymbol{W}_l \in \mathbb{R}^{d_l^{\text{out}} \times d_l^{\text{in}}}$ be a frozen target-model weight. Given an input c, a hypernetwork $H_\phi = D_\phi \circ E_\phi$ consists of a frozen encoder E_ϕ that maps c to a coordinate $\boldsymbol{z}$ and a frozen decoder D_ϕ that maps coordinates to LoRA factors:

$$\boldsymbol{z} = E_\phi(c) \in \mathbb{R}^p, \qquad D_\phi(\boldsymbol{z}) = \{(\boldsymbol{A}_l(\boldsymbol{z}), \boldsymbol{B}_l(\boldsymbol{z}))\}_{l \in \mathcal{L}}, \tag{1}$$

where $\boldsymbol{A}_l(\boldsymbol{z}) \in \mathbb{R}^{r \times d_l^{\text{in}}}$, $\boldsymbol{B}_l(\boldsymbol{z}) \in \mathbb{R}^{d_l^{\text{out}} \times r}$, r is the LoRA rank, and p is the dimension of the hypernetwork coordinate. The effective update and adapted weight are

$$\Delta \boldsymbol{W}_l(\boldsymbol{z}) = \alpha_l \boldsymbol{B}_l(\boldsymbol{z}) \boldsymbol{A}_l(\boldsymbol{z}), \qquad \boldsymbol{W}'_l(\boldsymbol{z}) = \boldsymbol{W}_l + \Delta \boldsymbol{W}_l(\boldsymbol{z}), \tag{2}$$

where α_l is a layer-specific scale. The encoder includes input preprocessing, and $\boldsymbol{z}$ may be a shared latent vector or vectorized layerwise latents.

Continual merging. Consider a stream of inputs $\{c_1, c_2, \ldots, c_T\}$. We use t to denote the current arrival and i to index an input among the first t arrivals. At arrival t, the hypernetwork produces coordinates $\boldsymbol{z}_t = E_\phi(c_t)$ and its corresponding LoRA. Our goal is to maintain a single merged representation $\boldsymbol{z}_m^{(t)}$ whose decoded LoRA approximates the functional effects of the LoRA observed so far. The desired merging objective is

$$\boldsymbol{z}_{\text{ideal}}^{(t)} = \arg\min_{\boldsymbol{z} \in \mathbb{R}^p} \sum\nolimits_{i=1}^{t} \mathcal{D}_i(\boldsymbol{z}, \boldsymbol{z}_i), \tag{3}$$

where $\mathcal{D}_i$ measures the discrepancy between the functional effects of the candidate LoRA decoded from $\boldsymbol{z}$ and generated latent vector $\boldsymbol{z}_i$. Section 4 will define this discrepancy and derive a tractable quadratic surrogate that supports incremental optimization. After each arrival, the merged coordinates $\boldsymbol{z}_m^{(t)}$ is decoded into a single LoRA, yielding deployed weights

$$\boldsymbol{W}_{l,t} = \boldsymbol{W}_l + \Delta \boldsymbol{W}_l(\boldsymbol{z}_m^{(t)}), \tag{4}$$

The merging processes each arrival once, retaining only the current merged representation and accumulated statistics. In our case, previous inputs and generated LoRAs are unavailable for revisit.

4 METHOD

In this section, we introduce our **MOSAIC**, which consolidates LoRAs in hypernetwork coordinates using a metric that connects these coordinates to their effects on the target model. The key idea is to measure disagreement through adapter-induced output changes and express this discrepancy in hypernetwork coordinates through the frozen decoder. This yields a matrix-weighted barycenter whose sufficient statistics can be accumulated online. We first define the functional matching objective, then derive its coordinate-space approximation and the resulting continual merging algorithm.

4.1 MATCHING ADAPTER BEHAVIOR

MOSAIC measures discrepancies between generated LoRAs through their effects on target-model activations. For incoming input c_t, the hypernetwork produces coordinates $\boldsymbol{z}_t = E_\phi(c_t)$ and effective updates $\Delta \boldsymbol{W}_l(\boldsymbol{z}_t)$. Processing c_t with the frozen target model provides input activations

$$\boldsymbol{X}_{t,l} = [x_{t,l,1}, \ldots, x_{t,l,n_t}] \in \mathbb{R}^{d_l^{\text{in}} \times n_t} \tag{5}$$

at each adapted layer $l \in \mathcal{L}$, where n_t is the number of valid token positions used as probes.

On these activations, the incoming LoRA contributes $\Delta \boldsymbol{W}_l(\boldsymbol{z}_t)\boldsymbol{X}_{t,l}$ to the layer output, whereas a candidate coordinate $\boldsymbol{z}$ contributes $\Delta \boldsymbol{W}_l(\boldsymbol{z})\boldsymbol{X}_{t,l}$. We define their functional discrepancy as

$$\mathcal{D}_t(\boldsymbol{z}, \boldsymbol{z}_t) = \frac{1}{2} \sum_{l \in \mathcal{L}} \left\| \left(\Delta \boldsymbol{W}_l(\boldsymbol{z}) - \Delta \boldsymbol{W}_l(\boldsymbol{z}_t)\right) \boldsymbol{X}_{t,l} \right\|_{\text{F}}^2 . \tag{6}$$

This specifies Eq. 3 and measures differences in LoRA output contributions on frozen-model activations, providing a proxy for functional preservation. The ideal merged coordinate minimizes the accumulated discrepancy:

$$F_t(\boldsymbol{z}) = \sum_{i=1}^{t} \mathcal{D}_i(\boldsymbol{z}, \boldsymbol{z}_i) \tag{7}$$

This objective compares layer-output contributions on frozen-model activations. Since the discrepancy sums over probe positions, inputs with more probes generally contribute more weight.

4.2 APPROXIMATING FUNCTIONAL DISCREPANCY IN HYPERNETWORK COORDINATE SPACE

Directly optimizing F_t would require repeatedly evaluating previous adapters on their probes, which conflicts with continual merging setting. To avoid this requirement, we approximate each incoming discrepancy by a quadratic function that can be retained via fixed-size statistics. Define $\boldsymbol{g}_t(\boldsymbol{z}) = \text{col}_{l \in \mathcal{L}}[\text{vec}(\Delta \boldsymbol{W}_l(\boldsymbol{z})\boldsymbol{X}_{t,l})]$ as the adapter's behavior on current probes across adapter layers. Let

$$\boldsymbol{G}_t = \left. \frac{\partial \boldsymbol{g}_t(\boldsymbol{z})}{\partial \boldsymbol{z}} \right|_{\boldsymbol{z}=\boldsymbol{z}_t} \tag{8}$$

be the Jacobian of the adapter's probe outputs. A first-order expansion gives

$$\boldsymbol{g}_t(\boldsymbol{z}) - \boldsymbol{g}_t(\boldsymbol{z}_t) \approx \boldsymbol{G}_t(\boldsymbol{z} - \boldsymbol{z}_t). \tag{9}$$

Substituting this approximation into the discrepancy yields

$$\mathcal{D}_t(\boldsymbol{z}, \boldsymbol{z}_t) \approx \frac{1}{2}(\boldsymbol{z} - \boldsymbol{z}_t)^\top \boldsymbol{M}_t(\boldsymbol{z} - \boldsymbol{z}_t), \quad \boldsymbol{M}_t = \boldsymbol{G}_t^\top \boldsymbol{G}_t. \tag{10}$$

The matrix $\boldsymbol{M}_t$ denotes which coordinate differences matter on the current probes. For any direction $\boldsymbol{u}$, $\boldsymbol{u}^\top \boldsymbol{M}_t \boldsymbol{u} = \|\boldsymbol{G}_t \boldsymbol{u}\|_2^2$, where a large value means that moving along $\boldsymbol{u}$ strongly changes adapter's probe outputs. We seek to minimize the resulting quadratic objective

$$Q_t(\boldsymbol{z}) = \frac{1}{2} \sum_{i=1}^{t} (\boldsymbol{z} - \boldsymbol{z}_i)^\top \boldsymbol{M}_i (\boldsymbol{z} - \boldsymbol{z}_i). \tag{11}$$

This is a matrix-weighted analogue of coordinate averaging. Each incoming adapter constrains the merged coordinate most strongly along directions that affect its measured behavior. Each $\boldsymbol{M}_i$ is constructed once at their respective coordinates $\boldsymbol{z}_i$ and then held fixed. The approximation is exact when the probe-output maps $\boldsymbol{g}_i$ are affine. For nonlinear decoders, its accuracy depends on decoder smoothness and distances between the candidate and incoming coordinates.

4.3 Online Merging with Sufficient Statistics

Although Eq. 11 sums over all previously observed coordinates, its minimizer depends only on two accumulated statistics:

$$\boldsymbol{S}_t = \sum_{i=1}^{t} \boldsymbol{M}_i, \quad \boldsymbol{b}_t = \sum_{i=1}^{t} \boldsymbol{M}_i \boldsymbol{z}_i \tag{12}$$

We expand the objective Eq. 11 as

$$Q_t(\boldsymbol{z}) = \frac{1}{2} \boldsymbol{z}^\top \boldsymbol{S}_t \boldsymbol{z} - \boldsymbol{z}^\top \boldsymbol{b}_t + \frac{1}{2} \sum_{i=1}^{t} \boldsymbol{z}_i^\top \boldsymbol{M}_i \boldsymbol{z}_i, \tag{13}$$

where the final term is independent of $\boldsymbol{z}$. Since each $\boldsymbol{M}_i$ is positive semidefinite, the minimizers satisfy $\boldsymbol{S}_t \boldsymbol{z} = \boldsymbol{b}_t$. The minimum-norm minimizer is therefore

$$\boldsymbol{z}_{\text{quad}}^{(t)} = \boldsymbol{S}_t^\dagger \boldsymbol{b}_t \tag{14}$$

where † denotes the Moore–Penrose pseudoinverse.

When a new input c_t arrives, we compute its coordinates $\boldsymbol{z}_t$ and metric $\boldsymbol{M}_t$, then update

$$\boldsymbol{S}_t = \boldsymbol{S}_{t-1} + \boldsymbol{M}_t, \quad \boldsymbol{b}_t = \boldsymbol{b}_{t-1} + \boldsymbol{M}_t \boldsymbol{z}_t, \quad N_t = N_{t-1} + n_t, \tag{15}$$

initialized with $\boldsymbol{S}_0 = \boldsymbol{0}_{p \times p}$, $\boldsymbol{b}_0 = \boldsymbol{0}_p$, and $N_0 = 0$. Here n_t is the probe-token count for the new input, and N_t is retained for metric stabilization. The accumulated statistics are invariant to permutations of incoming coordinates, metrics, and probe counts. Thus, the merged solution is order-invariant in exact arithmetic when probe selection and metric construction are independent of the preceding merged state.

Stabilization and decoding. To improve numerical conditioning, we track the total probe-token count N_t, normalize the accumulated matrix, and apply isotropic shrinkage:

$$\boldsymbol{\Sigma}_t = \frac{\boldsymbol{S}_t}{N_t}, \qquad \widetilde{\boldsymbol{\Sigma}}_t = (1 - \rho_t) \boldsymbol{\Sigma}_t + \rho_t \frac{\operatorname{tr}(\boldsymbol{\Sigma}_t)}{p} \boldsymbol{I}. \tag{16}$$

Here, $\rho_t \in [0, 1]$ is selected by OAS (Chen et al., 2010). The stabilized merged coordinate is

$$\boldsymbol{z}_m^{(t)} = \widetilde{\boldsymbol{\Sigma}}_t^\dagger \left(\frac{\boldsymbol{b}_t}{N_t} \right), \tag{17}$$

We apply shrinkage only when solving for the current coordinate, leaving $\boldsymbol{S}_t$ and $\boldsymbol{b}_t$ unchanged for subsequent arrivals. Decoding $\boldsymbol{z}_m^{(t)}$ produces the current merged LoRA, with deployed weights $\boldsymbol{W}_{l,t} = \boldsymbol{W}_l + \Delta \boldsymbol{W}_l(\boldsymbol{z}_m^{(t)})$. The completed algorithm is summarized in Algorithm 1.

5 Theoretical Analysis

In this section, we analyze: (1) whether sequential processing changes the solution of the accumulated quadratic objective, and (2) how accurately that objective approximates the nonlinear functional discrepancy. Proofs and additional stabilization analysis appear in Appendix A.

Algorithm 1 MOSAIC: Continual Merging in Hypernetwork Coordinates

Require: Input stream $\{c_t\}_{t\geq 1}$, frozen target model, encoder E_ϕ, decoder D_ϕ, adapter layers $\mathcal{L}$.
Ensure: A merged LoRA after each arrival.

```
Initialize (S_0, b_0, N_0) as (0_{p×p}, 0_p, 0).
for t = 1, 2, ... do
    Receive c_t and encode z_t ← E_φ(c_t).
    Collect frozen-model activations {X_{t,l}}_{l∈L} at n_t > 0 valid token positions.
    Define the current probe-output map g_t(z) = col_{l∈L}[vec(ΔW_l(z) X_{t,l})].
    Evaluate the probe-output Jacobian G_t using Eq. 8.
    Compute M_t ← G_t^⊤ G_t.
    Update (S_t, b_t, N_t) using Eq. 15
    Normalize Σ_t ← S_t/N_t , select ρ_t ← OAS(Σ_t, N_t), and form Σ̃_t using Eq. 16.
    Compute the coordinate z_m^(t) using Eq. 17.
    Decode D_φ(z_m^(t)) as the current merged LoRA.
    Retain (S_t, b_t, N_t).
end for
```

5.1 Order Invariance of Statistical Accumulation

We first compare sequential accumulation with solving the same quadratic problem using all observed adapters at once. Let $\boldsymbol{z}^{(t)}_{\text{quad}} = \boldsymbol{S}_t^\dagger \boldsymbol{b}_t$ denote the minimum-norm minimizer of the unregularized quadratic objective Q_t, distinguished from stabilized coordinate $\boldsymbol{z}_m^{(t)}$ produced by MOSAIC.

Proposition 1 (Order-Invariant Accumulation)**.** *For fixed incoming terms $\{(\boldsymbol{z}_i, \boldsymbol{M}_i, n_i)\}_{i=1}^t$ with $\boldsymbol{M}_i \succeq 0$, sequential and joint accumulation produce identical statistics $(\boldsymbol{S}_t, \boldsymbol{b}_t, N_t)$. The minimum-norm minimizer of Q_t is $\boldsymbol{z}^{(t)}_{\text{quad}} = \boldsymbol{S}_t^\dagger \boldsymbol{b}_t$. Both this minimizer and the stabilized solution $\boldsymbol{z}_m^{(t)}$ are independent of arrival order, provided the shrinkage coefficient is a deterministic function of $(\boldsymbol{S}_t, N_t)$.*

5.2 Bounding the Decoder Linearization Error

We next examine the approximation used to construct Q_t. For each arrival t, let $\boldsymbol{g}_t$ and $\boldsymbol{G}_t$ denote the probe-output map and Jacobian defined in Section 4.2.

Proposition 2 (Decoder linearization error)**.** *For each $i \leq t$, assume $\nabla \boldsymbol{g}_i$ is L_i-Lipschitz in operator norm on the segment from $\boldsymbol{z}_i$ to $\boldsymbol{z}$. Let $K_i = \|\boldsymbol{G}_i\|_2$ and $r_i(\boldsymbol{z}) = \|\boldsymbol{z} - \boldsymbol{z}_i\|_2$. Then*

$$|F_t(\boldsymbol{z}) - Q_t(\boldsymbol{z})| \leq \varepsilon_t(\boldsymbol{z}), \text{ where } \varepsilon_t(\boldsymbol{z}) := \sum_{i=1}^{t} \left[\frac{K_i L_i}{2} r_i(\boldsymbol{z})^3 + \frac{L_i^2}{8} r_i(\boldsymbol{z})^4 \right]. \tag{18}$$

Effect of stabilization. Shrinkage modifies the quadratic objective while improving its conditioning. Define $\mu_t = \text{tr}(\boldsymbol{\Sigma}_t)/p$. Specifically, when $\rho_t > 0$ and $\mu_t > 0$, $\lambda_{\min}(\widetilde{\boldsymbol{\Sigma}}_t) \geq \rho_t \mu_t$, where ρ_t is the OAS shrinkage coefficient. For positive definite $\boldsymbol{\Sigma}_t$, shrinkage cannot increase its spectral condition number. Appendix A.5 establishes these properties and bounds the resulting change in the merged coordinate. These numerical guarantees do not imply improved downstream accuracy or statistical optimality of OAS for decoder-derived metrics.

6 Experiments

In this section, we evaluate MOSAIC in two hypernetwork LoRA generation settings: (1) document-conditioned LoRA generation with Doc-to-LoRA and (2) task-conditioned LoRA generation with Text-to-LoRA. Across multiple target LLMs and benchmarks, we examine its ability to consolidate generated adapters as document chunks or tasks arrive sequentially.

Table 1: Performance of document-conditioned LoRA merging across three target LLMs over chunk sizes of 512, 1024, and 2048 tokens. Each entry averages over QASPER, 2WikiMultihopQA, and MultiFieldQA. **Bold** and underlined mark the best and second-best available merging results.

	Gemma-2-2B			Qwen3-4B			Mistral-7B		
Method	512	1024	2048	512	1024	2048	512	1024	2048
C-Task Arithmetic	13.86	22.68	46.00	7.94	12.96	21.94	9.44	19.33	43.07
C-Average	59.70	60.16	60.89	40.06	40.62	41.04	53.01	52.72	54.76
C-TIES	26.87	41.03	51.31	11.36	16.47	27.68	16.87	33.68	50.43
C-KnOTS-TIES	14.47	24.17	45.69	8.06	13.00	22.42	9.26	23.50	45.44
S-Task Arithmetic	59.38	58.34	62.27	38.59	38.41	39.59	62.56	70.81	70.50
S-Average	61.28	63.04	61.96	40.09	42.59	42.93	69.87	72.70	**76.83**
S-TIES	58.32	61.02	61.95	38.94	41.10	42.66	64.73	69.59	73.67
S-KnOTS-TIES	58.26	62.81	60.94	39.66	39.23	40.54	62.48	64.29	64.97
DOP	48.97	58.33	62.33	37.82	38.36	40.96	57.87	60.72	66.17
OPCM	62.04	61.32	62.37	41.59	42.59	43.33	65.16	65.55	68.61
RegMean	**63.13**	**65.04**	62.30	40.42	40.34	41.83	68.94	**74.29**	72.94
MOSAIC (Ours)	62.67	63.04	**62.89**	**43.22**	**42.92**	**45.65**	**71.86**	72.38	76.26

6.1 Experimental Setup

Document-conditioned generation. We use Doc-to-LoRA (Charakorn et al., 2026) on three benchmarks from LongBench (Bai et al., 2024) (2WikiMultihopQA (Ho et al., 2020), MultiFieldQA (Bai et al., 2024), along with QASPER (Dasigi et al., 2021)), SQuAD (Rajpurkar et al., 2016), DROP (Dua et al., 2019), and ROPES (Lin et al., 2019) for reading comprehension, and LoCoMo (Maharana et al., 2024) for conversational memory. Target models are Gemma-2-2b-Instruct (Team et al., 2024), Qwen3-4B-Instruct-2507 (Yang et al., 2025), and Mistral-7B-Instruct-v0.2 (Jiang et al., 2023). We generate one LoRA per context chunk and merge LoRAs in document order, using chunk sizes of 128, 256, 512, 1024, 2048, 4096, and 8192 tokens.

Chunk size and number of adapters. Each context chunk produces one LoRA adapter, so the number of chunks determines the length of the merging stream. For a fixed context, smaller chunks produce more adapters to merge. Adapter counts also depend on context length and the target model's tokenization. Tab. 3 and Tab. 4 report the minimum and maximum numbers of generated adapters across documents for each dataset, model, and chunk size. All methods use the same number of adapters in each experimental setting.

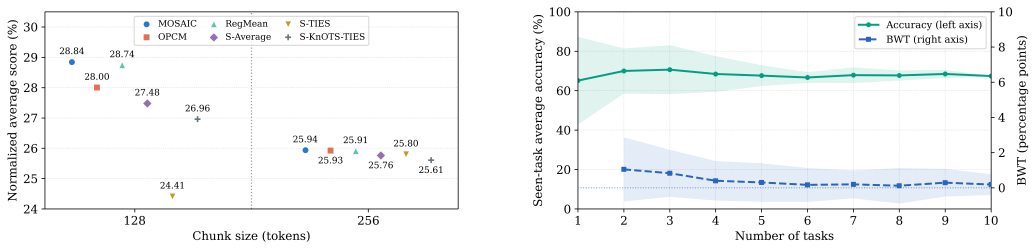
Task-conditioned generation. We use Text-to-LoRA (Charakorn et al., 2025) with Mistral-7B-Instruct-v0.2 and Qwen3-8B (Yang et al., 2025) on Arc-challenge (ArcC) and Arc-easy (ArcE) (Clark et al., 2018), BoolQ (Clark et al., 2019), GSM8K (Cobbe et al., 2021), Hellaswag (Zellers et al., 2019), OpenBookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020), Winogrande (Sakaguchi et al., 2021), HumanEval (Chen et al., 2021), and MBPP (Austin et al., 2021). We generate a LoRA per task and merge them in five random orders with distinct seeds.

Merging baselines. We compare MOSAIC with sequential implementations of weight averaging, Task Arithmetic (Ilharco et al., 2023), TIES-Merging (Yadav et al., 2023), and KnOTS (Stoica et al., 2025) with TIES. For each method, we evaluate two weighting schedules. **C-variants** (C-Task Arithmetic, C-Average, C-TIES, and C-KnOTS-TIES) use fixed weighting coefficients for adapters, while **S-variants** (S-Task Arithmetic, S-Average, S-TIES, and S-KnOTS-TIES) use arrival-dependent coefficients of $1/t$ at step t. We also include OPCM (Tang et al., 2025) and DOP (Yang et al., 2026), which are designed for continual model merging, and sequential implementation of RegMean (Jin et al., 2023). For document-conditioned generation, we use the **base model with context** as the upper bound, serving as the reference for normalizing all document-conditioned results. For task-conditioned generation, **Individual T2L** evaluates each task using its own hypernetwork-generated adapter. All experiments run on NVIDIA A100 GPUs.

Evaluation metrics. For document-conditioned LoRA generation, we evaluate ROUGE-L F1 normalized by the corresponding base model with context score. For task-conditioned LoRA genera-

Table 2: Performance of task-conditioned LoRA merging with Mistral-7B-Instruct-v0.2 and Qwen3-8B. Avg. reports the final ten-task macro-average (%), and BWT reports backward transfer. **Bold** and underlined mark the best and second-best means merging results.

	Mistral-7B		Qwen3-8B	
Method	Avg. ↑	BWT ↑	Avg. ↑	BWT ↑
Individual T2L	66.95	–	80.96	–
S-Task Arithmetic	60.34 ± 1.62	-3.37 ± 0.86	30.08 ± 4.62	-23.33 ± 2.94
S-Average	67.21 ± 0.31	$\underline{0.17} \pm 0.40$	82.33 ± 0.03	0.11 ± 0.38
S-TIES	66.31 ± 0.26	-0.42 ± 0.51	$\underline{84.02} \pm 0.23$	-0.9 ± 0.25
S-KnOTS-TIES	66.31 ± 0.18	-0.49 ± 0.49	70.00 ± 3.26	-3.86 ± 5.02
DOP	64.07 ± 0.21	-1.40 ± 0.50	81.16 ± 0.28	-0.16 ± 0.21
RegMean	65.14 ± 0.09	0.05 ± 0.39	82.28 ± 0.04	$\mathbf{0.28} \pm 0.12$
OPCM	$\underline{67.28} \pm 0.15$	0.12 ± 0.35	82.59 ± 0.10	0.08 ± 0.49
MOSAIC (Ours)	$\mathbf{67.44} \pm 0.08$	$\mathbf{0.21} \pm 0.52$	$\mathbf{84.37} \pm 0.12$	$\underline{0.24} \pm 0.12$



(a) Document-conditioned merging.

(b) Task-conditioned merging.

Figure 2: Document- and task-conditioned LoRA merging. (a) Comparison of merging methods on Qwen3-4B with 128- and 256-token chunks, averaged over SQuAD, DROP, and ROPES. (b) Seen-task average accuracy (left axis) and backward transfer (right axis) of MOSAIC on Mistral-7B across sequential arrivals. Lines show means across five task orders. BWT starts from second task.

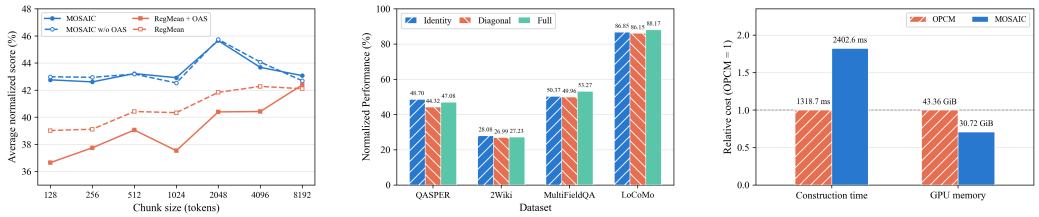
tion, we report accuracy on each benchmark except pass@1 on HumanEval and MBPP, and measure backward transfer as $\mathrm{BWT} = \frac{1}{T-1}\sum_{i=1}^{T-1}\left(R_{T,i} - R_{i,i}\right)$, where $R_{t,i}$ is performance on task i after merging t LoRAs. Positive BWT indicates improvement on earlier tasks.

6.2 Document-Conditioned LoRA Continual Merging

Tab. 1 compares MOSAIC with merging baselines on three different LLMs over chunk sizes of 512, 1024, and 2048 tokens. Chunks arrive in their original document order, and results average the normalized score over QASPER, 2WikiMultihopQA, and MultiFieldQA using the results of base model with context in Tab. 5. The S-variants consistently outperform their corresponding C-variants, suggesting that time-aware factor for LoRA is important for continual merging. MOSAIC outperforms OPCM and DOP at each chunk size across these three LLMs, and has better performance over RegMean in most cases. MOSAIC performs the best at each chunk size on Qwen3-4B and remains competitive on Gemma-2-2B and and Mistral-7B. Fig. 2a shows the comparison to SQuAD, DROP, and ROPES on Qwen3-4B using chunk sizes of 128 and 256. Results are also normalized by the results of base model with context in Tab. 5. MOSAIC outperforms other baselines at both chunk sizes, and most methods achieve higher scores with chunk size of 128. Together, these results indicate that smaller chunks and more adapters do not consistently improve performance.

6.3 Task-Conditioned LoRA Continual Merging

Tab. 2 compares task-conditioned LoRA merging on Mistral-7B and Qwen3-8B across five random task orders. MOSAIC achieves strong final macro-average performance on both models. It also surpasses Individual T2L, showing that consolidating generated LoRAs can improve average per-



(a) Impact of OAS. (b) Functional weighting. (c) Computational cost.

Figure 3: Ablations and computational cost. (a) Impact of OAS across chunk sizes, averaged over per-dataset normalized scores on QASPER, 2WikiMultihopQA, and MultiFieldQA. (b) Comparison of identity, diagonal, and full coordinate metrics using 1024-token chunks. (c) Construction time per arrival and peak additional GPU memory using 1024-token chunks, each normalized to OPCM.

formance over using a separate LoRA for each task. MOSAIC's variance in performance is due to generation sampling, and the generated LoRAs are identical regardless of order. Finally, MOSAIC achieves positive mean backward transfer on both models, indicating that subsequent merges improve performance on earlier tasks on average. Fig. 2b tracks performance of MOSAIC on Mistral-7B as the number of tasks increases from 1 to 10. Lines show the mean across five random task orders, and shaded bands indicate one standard deviation. Average accuracy initially increases and then remains relatively stable, while mean backward transfer generally decrease as more tasks arrive. The relatively narrow shaded bands and small standard deviations in Tab. 2 show that MOSAIC is robust to different task orderings.

6.4 Ablation Study

Impact of OAS. Fig. 3a compares MOSAIC and RegMean on Qwen3-4B across chunk sizes with and without OAS since both use activations during merging. MOSAIC achieves similar performance with and without OAS, with small changes depending on the chunk size. In contrast, RegMean performs worse with OAS. These results suggest that the effect of shrinkage depends on the merging formulation, with MOSAIC's coordinate-space formulation showing stable performance changes.

Impact of functional weighting. MOSAIC uses coordinate metric M_i to weight deviations from each incoming adapter's coordinates. This metric reflects layer-output changes induced by coordinate differences on frozen activation probes. Fig. 3b compares identity, diagonal, and full metrics using 1024-token chunks on Qwen3-4B. The identity metric weights all coordinate directions equally, the diagonal metric retains coordinate-specific weights, and the full metric additionally captures interactions between coordinates. The full metric improves the diagonal metric on all four datasets and achieves the highest scores on MultiFieldQA and LoCoMo. Identity performs best on QASPER and 2WikiMultihopQA, indicating that the benefit of coordinate interactions varies across datasets.

Computational Efficiency. Fig. 3c compares adapter construction costs on Qwen3-4B using 1024-token chunks. MOSAIC reduces peak additional GPU memory, while increasing construction time per arrival. These measurements concern construction costs rather than inference, and show that there is a trade-off between memory usage and adapter construction latency.

7 Conclusion

In this work, we introduced MOSAIC, a continual merging framework that consolidates hypernetwork-generated LoRAs in the hypernetwork's latent space. By pulling an activation-induced functional metric back through the frozen decoder, MOSAIC reduces merging to a matrix-weighted barycenter with additive sufficient statistics, enabling one-pass, order-invariant updates with memory independent of stream length and a single deployable LoRA after every arrival. Across document and task-conditioned generation with four target LLMs, MOSAIC consistently outperforms dedicated continual merging methods and remains competitive with the strongest weight-space baselines.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the manuscript, implement methods, assist in the writing of proofs.

We have not used generative AI tools to assist with translation, clean and reformat dataset, support qualitative and thematic data analysis, interpret results, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, propose or design or provide feedback on research methodology or experiments, and Generate synthetic data sets are not applicable to this work.

Additionally, we used generative AI tools for create or modify scientific figures, brainstorming.

We have reviewed all AI-assisted work. We checked LLM-generated research ideas for potential plagiarism through an extensive literature survey, LLM-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

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.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 3119–3137, 2024.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Rujikorn Charakorn, Edoardo Cetin, Yujin Tang, and Robert Tjarko Lange. Text-to-loRA: Instant transformer adaption. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=zWskCdu3QA.

Rujikorn Charakorn, Edoardo Cetin, Shinnosuke Uesaka, and Robert Tjarko Lange. Doc-to-loRA: Learning to instantly internalize contexts. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=iW1oBBO72S.

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.

Yilun Chen, Ami Wiesel, Yonina C Eldar, and Alfred O Hero. Shrinkage algorithms for mmse covariance estimation. *IEEE transactions on signal processing*, 58(10):5016–5029, 2010.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, 2019.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

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.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4599–4610, 2021.

Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2368–2378, 2019.

David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=rkpACe1lx.

Yun He, Steven Zheng, Yi Tay, Jai Gupta, Yu Du, Vamsi Aribandi, Zhe Zhao, YaGuang Li, Zhao Chen, Donald Metzler, et al. Hyperprompt: Prompt-based task-conditioning of transformers. In *International conference on machine learning*, pp. 8678–8690. PMLR, 2022.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

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

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6t0Kwf8-jrj.

Hamish Ivison and Matthew E Peters. Hyperdecoders: Instance-specific decoders for multi-task nlp. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 1715–1730, 2022.

Yihang Jiang, Xiaoyang Li, Guangxu Zhu, Hang Li, Jing Deng, Kaifeng Han, Chao Shen, Qingjiang Shi, and Rui Zhang. 6g non-terrestrial networks enabled low-altitude economy: Opportunities and challenges. *arXiv preprint arXiv:2311.09047*, 2023.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FCnohuR6AnM.

Kevin Lin, Oyvind Tafjord, Peter Clark, and Matt Gardner. Reasoning over paragraph effects in situations. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pp. 58–62, 2019.

Lihong Lin and Haidong Kang. Unlocking the potential of continual model merging: An ODE perspective. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=YLL4QeBbDD.

Rabeeh Karimi Mahabadi, Sebastian Ruder, Mostafa Dehghani, and James Henderson. Parameter-efficient multi-task fine-tuning for transformers via shared hypernetworks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 565–576, 2021.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2381–2391, 2018.

Oleksiy Ostapenko, Zhan Su, Edoardo Ponti, Laurent Charlin, Nicolas Le Roux, Lucas Caccia, and Alessandro Sordoni. Towards modular LLMs by building and reusing a library of loRAs. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=0ZFWfeVsaD.

Jason Phang, Yi Mao, Pengcheng He, and Weizhu Chen. Hypertuning: Toward adapting large language models without back-propagation. In *International Conference on Machine Learning*, pp. 27854–27875. pmlr, 2023.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 2383–2392, 2016.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with SVD to tie the knots. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=67X93aZHII.

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. URL https://openreview.net/forum?id=rdGMyTPhui.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

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

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. TIES-merging: Resolving interference when merging models. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xtaX3WyCj1.

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.

Enneng Yang, Anke Tang, Li Shen, Guibing Guo, Xingwei Wang, Xiaochun Cao, and Jie Zhang. Continual model merging without data: Dual projections for balancing stability and plasticity. *Advances in Neural Information Processing Systems*, 38:39275–39305, 2026.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

Aofan Yu, Chenyu Zhou, Tianyi Xu, Zihan Guo, Rong Shan, Zhihui Fu, Jun Wang, Weiwen Liu, Yong Yu, Weinan Zhang, et al. Latentskill: From in-context textual skills to in-weight latent skills for llm agents. *arXiv preprint arXiv:2606.06087*, 2026.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 4791–4800, 2019.

Ziyu Zhao, Didi Zhu, Zexi Li, Jing Su, Xuwu Wang, Fei Wu, et al. Merging loras like playing lego: Pushing the modularity of lora to extremes through rank-wise clustering. In *International Conference on Learning Representations*, volume 2025, pp. 72896–72913, 2025.

A Proofs and Additional Theoretical Analysis

This section supplies the derivation and proofs supporting Sections 4 and 5. We first relate the probe-output Jacobian to the layerwise weight-update Jacobians. We then prove the order-invariance and linearization-error results, identify the objective solved after stabilization, and quantify how shrinkage changes its conditioning and solution. All statements concern the functional discrepancy on frozen-model activation probes.

A.1 Notation and Setup

For each input t, let $\boldsymbol{z}_t \in \mathbb{R}^p$ be its hypernetwork coordinate and $\boldsymbol{X}_{t,l} \in \mathbb{R}^{d_l^{\text{in}} \times n_t}$ contain the frozen-model input activations at $n_t > 0$ valid token positions in adapted layer $l \in \mathcal{L}$. Define

$$\begin{aligned} \boldsymbol{g}_t(\boldsymbol{z}) &= \mathrm{col}_{l\in\mathcal{L}}\left[\mathrm{vec}\big(\Delta\boldsymbol{W}_l(\boldsymbol{z})\boldsymbol{X}_{t,l}\big)\right], \\ \boldsymbol{G}_t &= \left.\frac{\partial \boldsymbol{g}_t(\boldsymbol{z})}{\partial \boldsymbol{z}}\right|_{\boldsymbol{z}=\boldsymbol{z}_t}, \\ \boldsymbol{M}_t &= \boldsymbol{G}_t^\top \boldsymbol{G}_t. \end{aligned} \tag{19}$$

Here, vec stacks matrix columns and col concatenates vectors across layers. Thus, $\boldsymbol{M}_t \in \mathbb{R}^{p\times p}$ is symmetric positive semidefinite. The probes are held fixed when taking derivatives. The Jacobian $\boldsymbol{G}_t$ measures changes in probe outputs. It is distinct from the Jacobian of the effective weight update,

$$\boldsymbol{J}_{t,l} = \left.\frac{\partial \mathrm{vec}(\Delta\boldsymbol{W}_l(\boldsymbol{z}))}{\partial \boldsymbol{z}}\right|_{\boldsymbol{z}=\boldsymbol{z}_t} \in \mathbb{R}^{(d_l^{\text{out}} d_l^{\text{in}})\times p}. \tag{20}$$

Using $\mathrm{vec}(\boldsymbol{R}\boldsymbol{X}) = (\boldsymbol{X}^\top \otimes \boldsymbol{I})\mathrm{vec}(\boldsymbol{R})$ and differentiating with the probes fixed gives

$$\boldsymbol{G}_t = \mathrm{col}_{l\in\mathcal{L}}\left[(\boldsymbol{X}_{t,l}^\top \otimes \boldsymbol{I}_{d_l^{\text{out}}})\boldsymbol{J}_{t,l}\right]. \tag{21}$$

The Gram matrix of a stacked matrix is the sum of its block Gram matrices. Consequently,

$$\boldsymbol{M}_t = \boldsymbol{G}_t^\top \boldsymbol{G}_t = \sum_{l\in\mathcal{L}} \boldsymbol{J}_{t,l}^\top (\boldsymbol{X}_{t,l}\boldsymbol{X}_{t,l}^\top \otimes \boldsymbol{I}_{d_l^{\text{out}}})\boldsymbol{J}_{t,l}. \tag{22}$$

Both expressions account for changes in both LoRA factors and their fixed scale, because they differentiate the effective update $\Delta\boldsymbol{W}_l(\boldsymbol{z}) = \alpha_l \boldsymbol{B}_l(\boldsymbol{z})\boldsymbol{A}_l(\boldsymbol{z})$.

Neither requires explicitly storing a full Jacobian or a Kronecker matrix. For example, the product $\boldsymbol{M}_t\boldsymbol{u} = \boldsymbol{G}_t^\top(\boldsymbol{G}_t\boldsymbol{u})$ can be evaluated through a Jacobian–vector product followed by a vector–Jacobian product. Applying these products to coordinate basis vectors constructs the full matrix when needed. These are equivalent ways to evaluate the stated metric.

Objectives and accumulated statistics. Define

$$\begin{aligned} \mathcal{D}_t(\boldsymbol{z}, \boldsymbol{z}_t) &= \frac{1}{2}\|\boldsymbol{g}_t(\boldsymbol{z}) - \boldsymbol{g}_t(\boldsymbol{z}_t)\|_2^2, \\ q_t(\boldsymbol{z}) &= \frac{1}{2}\|\boldsymbol{G}_t(\boldsymbol{z} - \boldsymbol{z}_t)\|_2^2. \end{aligned} \tag{23}$$

and $F_t = \sum_{i=1}^t \mathcal{D}_i$ and $Q_t = \sum_{i=1}^t q_i$, we have

$$\begin{aligned} Q_t(\boldsymbol{z}) &= \frac{1}{2}\sum_{i=1}^t (\boldsymbol{z} - \boldsymbol{z}_i)^\top \boldsymbol{M}_i(\boldsymbol{z} - \boldsymbol{z}_i), \\ \boldsymbol{S}_t &= \sum_{i=1}^t \boldsymbol{M}_i, \qquad \boldsymbol{b}_t = \sum_{i=1}^t \boldsymbol{M}_i\boldsymbol{z}_i, \qquad N_t = \sum_{i=1}^t n_i > 0. \end{aligned} \tag{24}$$

The minimizer of Q_t with the smallest Euclidean norm is $\boldsymbol{z}_{\text{quad}}^{(t)} = \boldsymbol{S}_t^\dagger \boldsymbol{b}_t$, where $\dagger$ denotes the Moore–Penrose pseudoinverse.

Stabilization. Define

$$\begin{aligned} \boldsymbol{\Sigma}_t &= \boldsymbol{S}_t/N_t, & \mu_t &= \operatorname{tr}(\boldsymbol{\Sigma}_t)/p, \\ \widetilde{\boldsymbol{\Sigma}}_t &= (1-\rho_t)\boldsymbol{\Sigma}_t + \rho_t \mu_t \boldsymbol{I}_p, & \boldsymbol{z}_m^{(t)} &= \widetilde{\boldsymbol{\Sigma}}_t^{\dagger}(\boldsymbol{b}_t/N_t), \end{aligned} \tag{25}$$

The scalar μ_t is the average eigenvalue of the normalized sensitivity matrix, and $\rho_t \in [0,1]$ controls shrinkage toward an isotropic matrix.

A.2 Proof of Order-Invariant Accumulation

Proof of Proposition 1. Expanding the objective gives

$$Q_t(\boldsymbol{z}) = \frac{1}{2}\boldsymbol{z}^\top \boldsymbol{S}_t \boldsymbol{z} - \boldsymbol{z}^\top \boldsymbol{b}_t + \frac{1}{2}\sum_{i=1}^{t} \boldsymbol{z}_i^\top \boldsymbol{M}_i \boldsymbol{z}_i. \tag{26}$$

Since $\boldsymbol{S}_t \succeq 0$, Q_t is convex, and its minimizers satisfy $\boldsymbol{S}_t \boldsymbol{z} = \boldsymbol{b}_t$. We first show that this system has a solution. For any $\boldsymbol{v}$ in the null space $\ker(\boldsymbol{S}_t)$,

$$0 = \boldsymbol{v}^\top \boldsymbol{S}_t \boldsymbol{v} = \sum_{i=1}^{t} \boldsymbol{v}^\top \boldsymbol{M}_i \boldsymbol{v} = \sum_{i=1}^{t} \|\boldsymbol{M}_i^{1/2}\boldsymbol{v}\|_2^2. \tag{27}$$

This implies $\boldsymbol{M}_i \boldsymbol{v} = 0$ for every i. Therefore $\boldsymbol{v}^\top \boldsymbol{b}_t = \sum_i \boldsymbol{v}^\top \boldsymbol{M}_i \boldsymbol{z}_i = 0$. Thus $\boldsymbol{b}_t \in \ker(\boldsymbol{S}_t)^\perp = \operatorname{range}(\boldsymbol{S}_t)$, so the linear system has a solution, and its minimizers are

$$\arg\min_{\boldsymbol{z}} Q_t(\boldsymbol{z}) = \boldsymbol{S}_t^{\dagger}\boldsymbol{b}_t + \ker(\boldsymbol{S}_t). \tag{28}$$

The pseudoinverse solution is orthogonal to $\ker(\boldsymbol{S}_t)$. Adding any nonzero vector from this null space increases its Euclidean norm, so $\boldsymbol{z}_{\text{quad}}^{(t)} = \boldsymbol{S}_t^{\dagger}\boldsymbol{b}_t$ is the unique minimum-norm minimizer.

Starting from $\boldsymbol{S}_0 = 0$, $\boldsymbol{b}_0 = 0$, and $N_0 = 0$, the updates in Eq. 15 are $\boldsymbol{S}_i = \boldsymbol{S}_{i-1} + \boldsymbol{M}_i$, $\boldsymbol{b}_i = \boldsymbol{b}_{i-1} + \boldsymbol{M}_i \boldsymbol{z}_i$, and $N_i = N_{i-1} + n_i$. For fixed incoming terms $\{(\boldsymbol{z}_i, \boldsymbol{M}_i, n_i)\}_{i=1}^t$, these sums are unchanged in exact arithmetic by any permutation of the same collection of arrivals. The resulting solution $\boldsymbol{z}_{\text{quad}}^{(t)}$ is therefore independent of arrival order. If ρ_t is a deterministic function of $(\boldsymbol{S}_t, N_t)$, Eq. 25 shows that $\boldsymbol{z}_m^{(t)}$ is also unchanged. □

A.3 Proof of the Decoder Linearization Error Bound

Let g_i and $\boldsymbol{G}_i = \nabla\nabla \boldsymbol{g}_i(\boldsymbol{z}_i)$ be defined as in Section 5.2. Define

$$q_i(\boldsymbol{z}) := \frac{1}{2}\|\boldsymbol{G}_i(\boldsymbol{z} - \boldsymbol{z}_i)\|_2^2. \tag{29}$$

Then $Q_t = \sum_{i=1}^t q_i$ since $\boldsymbol{M}_i = \boldsymbol{G}_i^\top \boldsymbol{G}_i$. The original functional objective is $F_t(\boldsymbol{z}) = \frac{1}{2}\sum_i \|\boldsymbol{g}_i(\boldsymbol{z}) - \nabla \boldsymbol{g}_i(\boldsymbol{z}_i)\|_2^2$.

Proof of Proposition 2. Fix i and let $\boldsymbol{d}_i = \boldsymbol{z} - \boldsymbol{z}_i$. Using $r_i(\boldsymbol{z}) = \|\boldsymbol{d}_i\|_2$ and $K_i = \|\boldsymbol{G}_i\|_2$, we have

$$\begin{aligned} \boldsymbol{g}_i(\boldsymbol{z}) - \boldsymbol{g}_i(\boldsymbol{z}_i) &= \boldsymbol{G}_i \boldsymbol{d}_i + \boldsymbol{e}_i(\boldsymbol{z}), \\ \boldsymbol{e}_i(\boldsymbol{z}) &= \int_0^1 [\nabla \boldsymbol{g}_i(\boldsymbol{z}_i + s\boldsymbol{d}_i) - \boldsymbol{G}_i]\boldsymbol{d}_i \, \mathrm{d}s. \end{aligned} \tag{30}$$

Here $\boldsymbol{e}_i(\boldsymbol{z})$ is the first-order approximation error. The Lipschitz assumption bounds this error by

$$\|\boldsymbol{e}_i(\boldsymbol{z})\|_2 \le \int_0^1 L_i s \|\boldsymbol{d}_i\|_2^2 \, \mathrm{d}s = \frac{L_i}{2} r_i(\boldsymbol{z})^2. \tag{31}$$

Expanding the squared norm yields

$$\mathcal{D}_i(\boldsymbol{z}, \boldsymbol{z}_i) - q_i(\boldsymbol{z}) = \langle \boldsymbol{G}_i \boldsymbol{d}_i, \boldsymbol{e}_i(\boldsymbol{z})\rangle + \frac{1}{2}\|\boldsymbol{e}_i(\boldsymbol{z})\|_2^2. \tag{32}$$

By Cauchy–Schwarz and $\|\boldsymbol{G}_i \boldsymbol{d}_i\|_2 \le K_i r_i(\boldsymbol{z})$, we obtain

$$|\mathcal{D}_i(\boldsymbol{z}, \boldsymbol{z}_i) - q_i(\boldsymbol{z})| \le \frac{K_i L_i}{2} r_i(\boldsymbol{z})^3 + \frac{L_i^2}{8} r_i(\boldsymbol{z})^4. \tag{33}$$

Summing over i and applying the triangle inequality proves Eq. 18. □

If the effective updates $\Delta \boldsymbol{W}_l(\boldsymbol{z})$ are affine in $\boldsymbol{z}$, the approximation errors vanish. For nonlinear effective updates, the bound depends on the distances to the incoming coordinates and the variation of the Jacobians along the corresponding segments. It applies to the functional discrepancy on frozen activation probes.

A.4 The Objective After Stabilization

Let $\boldsymbol{E}_t$ denote the change in the normalized metric caused by shrinkage:

$$\boldsymbol{E}_t = \widetilde{\boldsymbol{\Sigma}}_t - \boldsymbol{\Sigma}_t = \rho_t(\mu_t \boldsymbol{I}_p - \boldsymbol{\Sigma}_t). \tag{34}$$

Then $\boldsymbol{z}_m^{(t)}$ is the minimum-norm minimizer of

$$\widetilde{Q}_t(\boldsymbol{z}) = Q_t(\boldsymbol{z}) + \frac{N_t}{2} \boldsymbol{z}^\top \boldsymbol{E}_t \boldsymbol{z}. \tag{35}$$

Its Hessian is $\boldsymbol{S}_t + N_t \boldsymbol{E}_t = N_t \widetilde{\boldsymbol{\Sigma}}_t \succeq 0$, and its minimizers satisfy $\widetilde{\boldsymbol{\Sigma}}_t \boldsymbol{z} = \boldsymbol{b}_t / N_t$. (1) If $\rho_t = 0$, existence follows from Proposition 1. (2) If $\rho_t > 0$ and $\mu_t > 0$, the matrix is positive definite. And (3) if $\mu_t = 0$, positive semidefiniteness implies $\boldsymbol{S}_t = 0$ and hence $\boldsymbol{b}_t = 0$. Its minimum-norm solution is therefore $\boldsymbol{z}_m^{(t)}$, as defined in Eq. 25.

Although $\widetilde{Q}_t$ is convex, $\boldsymbol{E}_t$ can have negative eigenvalues. So shrinkage changes the quadratic weighting and is not generally equivalent to adding a nonnegative penalty to Q_t.

A.5 Effect of Shrinkage on Conditioning and the Merged Representation

Let $\lambda_1, \ldots, \lambda_p$ be the eigenvalues of $\boldsymbol{\Sigma}_t$, with smallest and largest values $\lambda_{\min}$ and $\lambda_{\max}$. For a positive definite matrix $\boldsymbol{A}$, its spectral condition number is $\kappa_2(\boldsymbol{A}) = \lambda_{\max}(\boldsymbol{A})/\lambda_{\min}(\boldsymbol{A})$. The matrices $\boldsymbol{\Sigma}_t$ and $\widetilde{\boldsymbol{\Sigma}}_t$ share an orthonormal eigenbasis, with

$$\widetilde{\lambda}_j = (1 - \rho_t)\lambda_j + \rho_t \mu_t. \tag{36}$$

If $\rho_t > 0$ and $\mu_t > 0$, then $\widetilde{\boldsymbol{\Sigma}}_t \succ 0$ and $\lambda_{\min}(\widetilde{\boldsymbol{\Sigma}}_t) \geq \rho_t \mu_t$. If $\boldsymbol{\Sigma}_t \succ 0$, then

$$\kappa_2(\widetilde{\boldsymbol{\Sigma}}_t) = \frac{(1 - \rho_t)\lambda_{\max} + \rho_t \mu_t}{(1 - \rho_t)\lambda_{\min} + \rho_t \mu_t} \leq \kappa_2(\boldsymbol{\Sigma}_t). \tag{37}$$

Thus, positive shrinkage makes a nonzero singular metric invertible and does not increase the condition number of a positive definite metric.

Change in the merged representation. Define $\boldsymbol{\delta}_t = \boldsymbol{z}_m^{(t)} - \boldsymbol{z}_{\text{quad}}^{(t)}$. If $\widetilde{\boldsymbol{\Sigma}}_t \succ 0$, subtracting $\boldsymbol{\Sigma}_t \boldsymbol{z}_{\text{quad}}^{(t)} = \boldsymbol{b}_t / N_t$ from $\widetilde{\boldsymbol{\Sigma}}_t \boldsymbol{z}_m^{(t)} = \boldsymbol{b}_t / N_t$ gives $\widetilde{\boldsymbol{\Sigma}}_t \boldsymbol{\delta}_t = -\boldsymbol{E}_t \boldsymbol{z}_{\text{quad}}^{(t)}$. Hence

$$\boldsymbol{\delta}_t = -\widetilde{\boldsymbol{\Sigma}}_t^{-1} \boldsymbol{E}_t \boldsymbol{z}_{\text{quad}}^{(t)}. \tag{38}$$

This identity also holds when $\boldsymbol{\Sigma}_t$ is singular, since $\boldsymbol{\Sigma}_t \boldsymbol{z}_{\text{quad}}^{(t)} = \boldsymbol{b}_t / N_t$ remains valid. Therefore,

$$\|\boldsymbol{\delta}_t\|_2 \leq \frac{\rho_t \|\mu_t \boldsymbol{I}_p - \boldsymbol{\Sigma}_t\|_2}{\lambda_{\min}(\widetilde{\boldsymbol{\Sigma}}_t)} \|\boldsymbol{z}_{\text{quad}}^{(t)}\|_2. \tag{39}$$

Since $\boldsymbol{S}_t \boldsymbol{z}_{\text{quad}}^{(t)} = \boldsymbol{b}_t$, expanding around the quadratic minimizer gives

$$Q_t(\boldsymbol{z}_m^{(t)}) - Q_t(\boldsymbol{z}_{\text{quad}}^{(t)}) = \frac{1}{2} \boldsymbol{\delta}_t^\top \boldsymbol{S}_t \boldsymbol{\delta}_t \leq \frac{N_t \lambda_{\max}(\boldsymbol{\Sigma}_t)}{2} \|\boldsymbol{\delta}_t\|_2^2. \tag{40}$$

Combining this with Eq. 39 yields

$$0 \leq Q_t(\boldsymbol{z}_m^{(t)}) - Q_t(\boldsymbol{z}_{\text{quad}}^{(t)}) \leq \frac{N_t \lambda_{\max}(\boldsymbol{\Sigma}_t)}{2} \left(\frac{\rho_t \|\mu_t \boldsymbol{I}_p - \boldsymbol{\Sigma}_t\|_2}{\lambda_{\min}(\widetilde{\boldsymbol{\Sigma}}_t)} \right)^2 \|\boldsymbol{z}_{\text{quad}}^{(t)}\|_2^2. \tag{41}$$

These results quantify the effect of shrinkage on conditioning and on the quadratic solution.

B Details on Experiments

B.1 Baseline Methods

We compare MOSAIC with weight-space merging methods adapted to sequentially arriving LoRA adapters. We describe the operations for a single adapted layer and omit the layer index. Let $\boldsymbol{W}_0$ denote the frozen base weight and $\boldsymbol{\Delta}_t = \boldsymbol{B}_t\boldsymbol{A}_t$ denote the update generated for input t, with the LoRA scaling factor. The merged LoRA update is denoted by $\boldsymbol{W}^{(t)}_{\text{merged}}$, with

$$\boldsymbol{W}^{(0)}_{\text{merged}} = \boldsymbol{0}. \tag{42}$$

Thus, the resulting model weight is $\boldsymbol{W}_0 + \boldsymbol{W}^{(t)}_{\text{merged}}$.

Task Arithmetic. Task Arithmetic combines task-specific parameter updates with a shared pre-trained model. Applied to generated LoRAs, the merged update is given by

$$\boldsymbol{W}_{\text{merged}} = \lambda \sum_{i=1}^{T} \boldsymbol{\Delta}_i = \lambda \sum_{i=1}^{T} \boldsymbol{B}_i\boldsymbol{A}_i, \tag{43}$$

where λ is a scaling factor. We consider two continual variants. C-Task Arithmetic uses unit scaling, whereas S-Task Arithmetic scales the incoming update by the inverse arrival index:

$$\text{C-Task Arithmetic:} \quad \boldsymbol{W}^{(t)}_{\text{merged}} = \boldsymbol{W}^{(t-1)}_{\text{merged}} + \boldsymbol{\Delta}_t, \tag{44}$$

$$\text{S-Task Arithmetic:} \quad \boldsymbol{W}^{(t)}_{\text{merged}} = \boldsymbol{W}^{(t-1)}_{\text{merged}} + \frac{1}{t}\boldsymbol{\Delta}_t. \tag{45}$$

Expanding these recurrences gives $\sum_{i=1}^{t} \boldsymbol{\Delta}_i$ for C-Task Arithmetic and $\sum_{i=1}^{t} \boldsymbol{\Delta}_i/i$ for S-Task Arithmetic.

Weight Averaging. Weight averaging combines model parameters through a weighted mean. In the LoRA setting, we apply this operation to the generated weight updates. C-Average gives equal weight to the current merged update and the incoming update. S-Average accounts for the number of adapters already observed:

$$\text{C-Average:} \quad \boldsymbol{W}^{(t)}_{\text{merged}} = \frac{\boldsymbol{W}^{(t-1)}_{\text{merged}} + \boldsymbol{\Delta}_t}{2}, \tag{46}$$

$$\text{S-Average:} \quad \boldsymbol{W}^{(t)}_{\text{merged}} = \frac{(t-1)\boldsymbol{W}^{(t-1)}_{\text{merged}} + \boldsymbol{\Delta}_t}{t}. \tag{47}$$

Both recurrences start from $\boldsymbol{W}^{(1)}_{\text{merged}} = \boldsymbol{\Delta}_1$ and apply for $t \geq 2$. C-Average emphasizes recent arrivals, while S-Average gives

$$\boldsymbol{W}^{(t)}_{\text{merged}} = \frac{1}{t} \sum_{i=1}^{t} \boldsymbol{\Delta}_i. \tag{48}$$

This is the running-average rule used in stochastic weight averaging.

TIES-Merging. TIES-Merging addresses interference caused by small parameter updates and conflicting signs. It first removes small-magnitude entries, selects a sign for each coordinate, and retains the entries that agree with the selected sign. Write

$$\big(\boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_T\big) = \text{TIESSelect}_k\big(\boldsymbol{\Delta}_1, \ldots, \boldsymbol{\Delta}_T; w_1, \ldots, w_T\big), \tag{49}$$

where k is retained fraction of entries and $\boldsymbol{\tau}_i$ is the trimmed, sign-consistent update from model i.

In continual merging, the two LoRAs are the previous merged update and the incoming LoRA. For positive weights a, b, define

$$\big(\boldsymbol{\tau}^{a,b}_{\text{prev}}, \boldsymbol{\tau}^{a,b}_{\text{new}}\big) = \text{TIESSelect}_k\big(\boldsymbol{W}^{(t-1)}_{\text{merged}}, \boldsymbol{\Delta}_t; a, b\big). \tag{50}$$

C-TIES treats the two LoRAs equally and sums the selected entries:

$$\text{C-TIES:} \quad \boldsymbol{W}_{\text{merged}}^{(t)} = \lambda_{\text{TIES}} \left(\boldsymbol{\tau}_{\text{prev}}^{1,1} + \boldsymbol{\tau}_{\text{new}}^{1,1} \right). \tag{51}$$

S-TIES uses weights $t-1$ and 1 in both sign selection and disjoint averaging. At coordinate j, its update is

$$\text{S-TIES:} \quad \left[\boldsymbol{W}_{\text{merged}}^{(t)}\right]_j = \frac{(t-1)[\boldsymbol{\tau}_{\text{prev}}^{t-1,1}]_j + [\boldsymbol{\tau}_{\text{new}}^{t-1,1}]_j}{(t-1)\mathbb{I}\{[\boldsymbol{\tau}_{\text{prev}}^{t-1,1}]_j \neq 0\} + \mathbb{I}\{[\boldsymbol{\tau}_{\text{new}}^{t-1,1}]_j \neq 0\}}. \tag{52}$$

The denominator counts LoRA whose entries survive sign selection.

KnOTS-TIES. KnOTS aligns LoRA updates in a shared space before applying a merging rule. Our sequential adaptation constructs common output and input bases for the previous merged update and the incoming LoRA. Let $\boldsymbol{U}_t$ and $\boldsymbol{V}_t$ be orthonormal bases spanning the columns of $[\boldsymbol{B}_{t-1}^m, \boldsymbol{B}_t]$ and $[(\boldsymbol{A}_{t-1}^m)^\top, \boldsymbol{A}_t^\top]$, respectively, where m denotes the previously merged LoRA factors. The updates in the shared space are

$$\boldsymbol{P}_t = \boldsymbol{U}_t^\top \boldsymbol{W}_{\text{merged}}^{(t-1)} \boldsymbol{V}_t, \qquad \boldsymbol{Q}_t = \boldsymbol{U}_t^\top \boldsymbol{\Delta}_t \boldsymbol{V}_t. \tag{53}$$

Let C-TIES denote the equal-weight selection and scaled sum defined above, and let S-TIES$_t$ denote the history-weighted selection and disjoint mean. The corresponding KnOTS-TIES are

$$\text{C-KnOTS-TIES:} \quad \boldsymbol{W}_{\text{merged}}^{(t)} = \boldsymbol{U}_t \text{C-TIES}(\boldsymbol{P}_t, \boldsymbol{Q}_t) \boldsymbol{V}_t^\top, \tag{54}$$

$$\text{S-KnOTS-TIES:} \quad \boldsymbol{W}_{\text{merged}}^{(t)} = \boldsymbol{U}_t \text{S-TIES}_t(\boldsymbol{P}_t, \boldsymbol{Q}_t) \boldsymbol{V}_t^\top. \tag{55}$$

RegMean. RegMean merges linear weights by minimizing discrepancies between the outputs of the merged model and outputs of the individual models. Let $\boldsymbol{X}_i \in \mathbb{R}^{d_{\text{in}} \times n_i}$ contain the input activation probes with adapter i, and define $\boldsymbol{G}_i = \boldsymbol{X}_i \boldsymbol{X}_i^\top$. RegMean preserves diagonal entries and multiplies off-diagonal entries by $\gamma = 0.9$:

$$\widehat{\boldsymbol{G}}_i = \gamma \boldsymbol{G}_i + (1-\gamma) \operatorname{Diag}\big(\operatorname{diag}(\boldsymbol{G}_i)\big). \tag{56}$$

The merged update minimizes the quadratic objective

$$\min_{\boldsymbol{W}} \frac{1}{2} \sum_{i=1}^{t} \operatorname{tr}\left[(\boldsymbol{W} - \boldsymbol{\Delta}_i) \widehat{\boldsymbol{G}}_i (\boldsymbol{W} - \boldsymbol{\Delta}_i)^\top \right]. \tag{57}$$

Its minimizer can be computed by two accumulated statistics, initialized as $\boldsymbol{H}_0 = \boldsymbol{0}$ and $\boldsymbol{R}_0 = \boldsymbol{0}$:

$$\begin{aligned} \boldsymbol{H}_t &= \boldsymbol{H}_{t-1} + \widehat{\boldsymbol{G}}_t, \\ \boldsymbol{R}_t &= \boldsymbol{R}_{t-1} + \boldsymbol{\Delta}_t \widehat{\boldsymbol{G}}_t. \end{aligned} \tag{58}$$

The minimum-Frobenius-norm solution is

$$\boldsymbol{W}_{\text{merged}}^{(t)} = \boldsymbol{R}_t \boldsymbol{H}_t^\dagger, \tag{59}$$

where † denotes the Moore–Penrose pseudoinverse. The statistics are retained between arrivals, so earlier adapters and activation batches are not used for later updates.

DOP. Dual Orthogonal Projection (DOP) merges updates by preserving important singular directions of both the previous merged model and the incoming model. Let $\boldsymbol{W}_o = \boldsymbol{W}_{\text{merged}}^{(t-1)}$ and $\boldsymbol{W}_n = \boldsymbol{\Delta}_t$. For each $j \in \{o, n\}$, let $(\boldsymbol{U}_j, \boldsymbol{\Lambda}_j, \boldsymbol{V}_j)$ denote the retained singular components of $\boldsymbol{W}_j$. The projection loss for a candidate update $\boldsymbol{W}$ is

$$\mathcal{L}_j(\boldsymbol{W}) = \|\boldsymbol{\Lambda}_j \boldsymbol{U}_j^\top (\boldsymbol{W} - \boldsymbol{W}_j)\|_F^2 + \|(\boldsymbol{W} - \boldsymbol{W}_j) \boldsymbol{V}_j \boldsymbol{\Lambda}_j\|_F^2. \tag{60}$$

Starting from $(\boldsymbol{W}_o + \boldsymbol{W}_n)/2$, DOP iteratively combines the gradients of the two losses using adaptive weights. The optimized update becomes $\boldsymbol{W}_{\text{merged}}^{(t)}$. The optimization uses the model updates and does not require labeled task examples.

Table 3: Minimum and maximum chunk counts across documents in the three LongBench datasets. Counts use each model's tokenizer and context-formatting template. Each chunk produces one LoRA adapter. We abbreviate 2WikiMultihopQA as 2WikiMQA.

Model	Chunk size	QASPER		2WikiMQA		MultiFieldQA	
		Min	Max	Min	Max	Min	Max
Gemma-2-2B	128	15	168	12	133	10	129
	256	8	84	6	67	5	65
	512	4	42	3	34	3	33
	1024	2	21	2	17	2	17
	2048	1	11	1	9	1	9
	4096	1	6	1	5	1	5
	8192	1	3	1	3	1	3
Qwen3-4B	128	16	172	12	135	11	129
	256	8	86	6	68	6	65
	512	4	43	3	34	3	33
	1024	2	22	2	17	2	17
	2048	1	11	1	9	1	9
	4096	1	6	1	5	1	5
	8192	1	3	1	3	1	3
Mistral-7B	128	17	189	13	150	11	139
	256	9	95	7	75	6	70
	512	5	48	4	38	3	35
	1024	3	24	2	19	2	18
	2048	2	12	1	10	1	9
	4096	1	6	1	5	1	5
	8192	1	3	1	3	1	3

OPCM. Orthogonal Projection-based Continual Merging (OPCM) projects the incoming update away from selected directions of the previous merged update. Let $\boldsymbol{U}_t$ and $\boldsymbol{V}_t$ contain the selected left and right singular vectors of the retained state. The projection is

$$\mathcal{P}_{t-1}(\boldsymbol{\Delta}_t) = \boldsymbol{\Delta}_t - \boldsymbol{U}_t \boldsymbol{U}_t^\top \boldsymbol{\Delta}_t \boldsymbol{V}_t \boldsymbol{V}_t^\top. \tag{61}$$

The projected update is combined with the previous state using

$$\boldsymbol{W}_{\text{merged}}^{(t)} = \frac{\lambda^{(t-1)} \boldsymbol{W}_{\text{merged}}^{(t-1)} + \mathcal{P}_{t-1}(\boldsymbol{\Delta}_t)}{\lambda^{(t)}}. \tag{62}$$

The recurrence starts from $\boldsymbol{W}_{\text{merged}}^{(1)} = \boldsymbol{\Delta}_1$. The singular vectors are selected according to the singular-value threshold.

Base model with context. For document-conditioned evaluation, we provide the document context and question directly to the frozen target LLM without a generated adapter. This baseline measures performance when the source information is available in the input and serves as the reference for normalizing all document-conditioned results.

Base model. For task-conditioned evaluation, we evaluate the target LLM without a generated adapter under the same task evaluation protocol. This baseline provides a reference for assessing the effect of adapter generation and merging.

Individual T2L. For task-conditioned evaluation, we evaluate each task with the adapter generated from its own task description and report the average task score. No merging is performed. This baseline uses a separate adapter for each task, whereas the merged model uses a single consolidated adapter across tasks.

B.2 Number of Chunks/LoRAs

We split each document into token-based chunks, with each chunk producing one LoRA adapter. For a fixed document, tokenizer, and chunk size, the number of chunks is deterministic. However,

Table 4: Maximum chunk counts across evaluation examples in SQuAD, DROP, and ROPES. The minimum is 1 for every setting and is omitted. The final row of each model block applies to all three tested chunk sizes: 2048, 4096, and 8192.

Model	Chunk size	SQuAD	DROP	ROPES
Gemma-2-2B	128	6	11	3
	256	3	6	2
	512	2	3	1
	1024	1	2	1
	2048–8192	1	1	1
Qwen3-4B	128	7	11	3
	256	4	6	2
	512	2	3	1
	1024	1	2	1
	2048–8192	1	1	1
Mistral-7B	128	7	13	4
	256	4	7	2
	512	2	4	1
	1024	1	2	1
	2048–8192	1	1	1

Table 5: Performance of base model with context in document-conditioned LoRA setting. Scores serve as the per-dataset references for normalizing document-conditioned LoRA merging results.

Model	QASPER	2WikiMultihopQA	MultiFieldQA	LoCoMo	SQuAD	DROP	ROPES
Gemma-2-2B	38.72	33.46	39.59	19.20	87.51	42.60	72.74
Qwen3-4B	45.09	38.11	47.58	12.49	77.75	45.08	70.82
Mistral-7B	27.55	22.36	46.04	16.46	51.47	23.13	11.83

it varies across documents because their lengths differ, and across models because their tokenizers and context-formatting templates can produce different token lengths for the same text. Tab. 3 and Tab. 4 therefore report the minimum and maximum chunk counts across documents in each dataset for each model and chunk size. Smaller chunk sizes generally produce more chunks, resulting in longer sequences of adapters to merge. We used the same seed to keep each baseline meet same number of chunks with each other.

B.3 Base model with context

Tab. 5 shows the absolute performance of each dataset across three LLMs in base model with context, which is viewed as the upper bound of the document-conditioned generation. These scores serve as the per-dataset references for normalizing document-conditioned LoRA merging results in Tab. 1, Tab. 6, Tab. 7, Tab. 8, Tab. 9, Fig. 3a and Fig. 3b.

B.4 Document-conditioned LoRA merging with different chunk sizes

Tables 6, 7, and 8 report LongBench results, Table 9 report LoCoMo results. LongBench results use the chunk sizes of 512, 1024, and 2048 tokens, while LoCoMo results 1024 tokens.

B.5 Text-conditioned LoRA merging

Tab. 11, Tab 12, and Tab. 10 show the specific result of each dataset across Qwen3-8B and Mistral-7B

Table 6: Document-conditioned LoRA merging on LongBench with 512-token chunks. Scores are normalized by the corresponding base-model-with-context scores and multiplied by 100. Avg. is the mean over the three datasets. 2Wiki denotes 2WikiMultihopQA. **Bold** and underlining indicate the best and second-best distinct scores within each model, including ties at the displayed precision.

Method	Avg.	QASPER	2Wiki	MultiFieldQA
Gemma-2-2B				
C-Task Arithmetic	13.86	11.19	11.76	18.65
C-TIES	26.87	24.30	32.90	23.41
C-KnOTS-TIES	14.47	11.93	14.02	17.48
S-Average	61.28	47.15	**95.43**	41.27
S-TIES	58.32	47.06	84.16	43.73
S-KnOTS-TIES	58.26	50.80	84.59	39.38
OPCM	62.04	49.71	92.03	44.38
RegMean	**63.13**	**51.04**	92.11	**46.25**
MOSAIC (Ours)	62.67	49.62	94.49	43.88
Qwen3-4B				
C-Task Arithmetic	7.94	4.55	1.89	17.38
C-TIES	11.36	8.04	4.53	21.52
C-KnOTS-TIES	8.06	4.82	1.74	17.61
S-Average	40.09	42.29	**31.88**	46.09
S-TIES	38.94	44.76	24.60	47.46
S-KnOTS-TIES	39.66	**46.40**	22.57	50.00
OPCM	41.59	46.11	28.64	50.01
RegMean	40.42	42.17	28.38	50.72
MOSAIC (Ours)	**43.22**	45.87	28.96	**54.82**
Mistral-7B				
C-Task Arithmetic	9.44	3.84	8.24	16.23
C-TIES	16.87	9.40	17.67	23.53
C-KnOTS-TIES	9.26	3.33	8.42	16.04
S-Average	69.87	60.78	93.20	55.64
S-TIES	64.73	52.49	84.24	57.45
S-KnOTS-TIES	62.48	50.78	82.06	54.59
OPCM	65.16	59.74	80.02	55.73
RegMean	68.94	59.86	89.05	**57.90**
MOSAIC (Ours)	**71.86**	**65.76**	**93.64**	56.19

Table 7: Document-conditioned LoRA merging on LongBench with 1024-token chunks. Scores are normalized by the corresponding base-model-with-context scores and multiplied by 100. Avg. is the mean over the three datasets. 2Wiki denotes 2WikiMultihopQA. **Bold** and underlining indicate the best and second-best distinct scores within each model, including ties at the displayed precision.

Method	Avg.	QASPER	2Wiki	MultiFieldQA
Gemma-2-2B				
C-Task Arithmetic	22.68	23.07	24.15	20.83
C-TIES	41.03	36.26	55.41	31.42
C-KnOTS-TIES	24.17	23.43	26.12	22.97
S-Average	63.04	49.10	93.65	46.37
S-TIES	61.02	49.61	92.46	41.00
S-KnOTS-TIES	62.81	**53.33**	89.62	45.47
OPCM	61.32	48.68	93.88	41.41
RegMean	**65.04**	50.66	**94.40**	**50.05**
MOSAIC (Ours)	63.04	49.69	93.03	46.42
Qwen3-4B				
C-Task Arithmetic	12.96	7.98	5.41	25.47
C-TIES	16.47	15.91	6.51	26.98
C-KnOTS-TIES	13.00	9.89	4.72	24.40
S-Average	42.59	**48.90**	28.41	50.47
S-TIES	41.10	48.13	22.48	52.67
S-KnOTS-TIES	39.23	44.97	21.14	51.59
OPCM	42.59	46.82	**29.38**	51.59
RegMean	40.34	45.23	26.19	49.59
MOSAIC (Ours)	**42.92**	46.88	29.10	**52.77**
Mistral-7B				
C-Task Arithmetic	19.33	12.41	22.17	23.41
C-TIES	33.68	26.39	41.72	32.94
C-KnOTS-TIES	23.50	14.98	29.79	25.72
S-Average	72.70	59.93	99.25	58.93
S-TIES	69.59	57.39	92.75	58.64
S-KnOTS-TIES	64.29	49.04	89.90	53.94
OPCM	65.55	55.18	85.72	55.76
RegMean	**74.29**	**60.39**	**101.54**	**60.93**
MOSAIC (Ours)	72.38	59.56	97.63	59.94

Table 8: Document-conditioned LoRA merging on LongBench with 2048-token chunks. Scores are normalized by the corresponding base-model-with-context scores and multiplied by 100. Avg. is the mean over the three datasets. 2Wiki denotes 2WikiMultihopQA. **Bold** and underlining indicate the best and second-best distinct scores within each model, including ties at the displayed precision.

Method	Avg.	QASPER	2Wiki	MultiFieldQA
Gemma-2-2B				
C-Task Arithmetic	46.00	38.80	58.05	41.16
C-TIES	51.31	41.41	67.49	45.04
C-KnOTS-TIES	45.69	41.55	55.60	39.92
S-Average	61.96	51.38	87.30	47.20
S-TIES	61.95	50.24	89.36	46.24
S-KnOTS-TIES	60.94	**54.23**	87.75	40.84
OPCM	62.37	50.87	**89.70**	46.53
RegMean	62.30	46.78	86.66	**53.44**
MOSAIC (Ours)	**62.89**	50.65	88.55	49.48
Qwen3-4B				
C-Task Arithmetic	21.94	26.64	9.15	30.03
C-TIES	27.68	32.49	11.49	39.05
C-KnOTS-TIES	22.42	26.22	9.89	31.14
S-Average	42.93	49.83	27.11	51.85
S-TIES	42.66	49.82	24.41	53.74
S-KnOTS-TIES	40.54	47.40	22.49	51.72
OPCM	43.33	48.81	28.79	52.40
RegMean	41.83	47.02	25.30	53.18
MOSAIC (Ours)	**45.65**	**51.90**	**30.05**	**55.01**
Mistral-7B				
C-Task Arithmetic	43.07	34.04	59.90	35.28
C-TIES	50.43	45.28	63.76	42.24
C-KnOTS-TIES	45.44	37.02	62.97	36.32
S-Average	**76.83**	64.04	**106.29**	**60.18**
S-TIES	73.67	62.00	100.21	58.79
S-KnOTS-TIES	64.97	50.94	88.51	55.46
OPCM	68.61	59.61	89.08	57.12
RegMean	72.94	59.83	99.40	59.58
MOSAIC (Ours)	76.26	**66.46**	105.15	57.17

Table 9: Document-conditioned LoRA merging on LoCoMo with 1024-token chunks. Scores are normalized by the corresponding base-model-with-context scores and multiplied by 100. **Bold** and underlining indicate the best and second-best distinct scores within each model, including ties at the displayed precision.

Method	LoCoMo
Gemma-2-2B	
C-Task Arithmetic	**65.24**
C-TIES	64.34
C-KnOTS-TIES	<u>64.90</u>
S-Average	63.07
S-TIES	62.45
S-KnOTS-TIES	64.36
OPCM	62.49
RegMean	63.68
MOSAIC (Ours)	63.53
Qwen3-4B	
C-Task Arithmetic	79.15
C-TIES	81.77
C-KnOTS-TIES	78.76
S-Average	86.41
S-TIES	84.62
S-KnOTS-TIES	86.24
OPCM	84.30
RegMean	**88.78**
MOSAIC (Ours)	<u>87.62</u>
Mistral-7B	
C-Task Arithmetic	71.53
C-TIES	71.62
C-KnOTS-TIES	71.96
S-Average	72.63
S-TIES	72.74
S-KnOTS-TIES	73.31
OPCM	73.76
RegMean	<u>74.34</u>
MOSAIC (Ours)	**74.53**

Table 10: Per-task final scores (%) on Mistral-7B, averaged over five task orders. Macro average reports the mean and sample standard deviation of the ten-task macro-average across orders. Bold marks the highest score among merging methods.

Method	ARC-C	ARC-E	BoolQ	GSM8K	HellaSwag	OBQA	PIQA	WinoGrande	HumanEval	MBPP	Macro average
Individual T2L	77.39	89.06	84.31	43.29	68.08	74.60	82.59	61.80	39.02	49.37	66.95
S-Task Arithmetic	75.20	87.95	83.82	25.56	63.63	74.32	**82.45**	61.99	14.02	34.44	60.34 ± 1.62
S-Average	<u>77.42</u>	88.67	84.76	<u>45.64</u>	65.74	<u>74.36</u>	82.35	61.90	39.51	51.78	67.21 ± 0.31
S-TIES	76.55	88.70	82.75	44.99	**66.58**	71.08	81.97	60.44	38.66	51.38	66.31 ± 0.26
S-KnOTS-TIES	76.81	**89.06**	**85.20**	44.47	66.18	74.24	81.48	<u>62.34</u>	34.80	48.52	66.31 ± 0.18
DOP	73.31	87.69	81.45	42.09	65.94	64.60	80.92	57.87	36.95	49.87	64.07 ± 0.21
RegMean	76.21	88.21	74.50	43.47	62.77	71.20	80.11	60.24	**42.32**	<u>52.43</u>	65.14 ± 0.09
OPCM	**77.49**	88.98	84.62	45.34	66.42	**74.60**	82.08	**62.94**	37.80	**52.53**	<u>67.28</u> ± 0.15
MOSAIC (Ours)	77.05	**89.06**	<u>85.11</u>	**46.17**	<u>66.45</u>	73.80	81.99	61.96	<u>40.37</u>	<u>52.43</u>	**67.44** ± 0.08

Table 11: Per-task final scores on Qwen3-8B, averaged over five orders. Average reports the ten-task mean ± standard deviation across orders. Bold marks the better result.

Method	ARC-C	ARC-E	BoolQ	GSM8K	HellaSwag	OBQA	PIQA	WinoGrande	HumanEval	MBPP	Macro	Micro
Individual T2L	90.78	95.75	85.87	72.33	81.63	85.20	85.26	69.85	70.73	72.18	80.96	83.14
OPCM	90.29	95.76	86.34	83.65	81.91	86.04	85.96	69.55	71.83	74.54	82.59 ± 0.10	84.09 ± 0.17
RegMean	90.92	96.20	85.96	80.02	82.25	86.80	86.72	69.16	70.61	74.19	82.28 ± 0.04	84.10 ± 0.01
DOP	89.95	95.67	86.60	79.85	81.16	86.36	85.51	66.39	68.05	72.08	81.16 ± 0.28	83.26 ± 0.21
S-Average	90.90	95.89	86.76	80.65	82.25	86.44	86.01	70.04	69.88	74.49	82.33 ± 0.03	84.20 ± 0.03
S-Task Arithmetic	30.65	36.51	60.49	0.15	26.39	39.72	53.85	51.73	1.34	0.00	30.08 ± 4.62	34.47 ± 3.95
S-TIES	91.23	96.34	86.48	**86.64**	82.61	86.52	**86.93**	68.51	**77.80**	**77.14**	84.02 ± 0.23	84.84 ± 0.24
S-KnOTS-TIES	86.52	93.51	84.05	26.85	73.66	80.64	81.25	65.37	49.63	58.55	70.00 ± 3.26	75.07 ± 1.95
MOSAIC (Ours)	**91.69**	**96.51**	**87.25**	86.58	**83.91**	**87.60**	86.61	**70.18**	77.20	76.19	**84.37 ± 0.12**	**85.64 ± 0.02**

Table 12: Final merged-LoRA scores (%) for Qwen3-8B, across five task orders. Each row evaluates the final ten-task adapter on all datasets. HumanEval and MBPP report base-test pass@1; other columns report accuracy. Bold marks the best score within each order and dataset, including ties at the displayed precision.

Method	ARC-C	ARC-E	BoolQ	GSM8K	HellaSwag	OBQA	PIQA	WinoGrande	HumanEval	MBPP
Order 1										
OPCM	90.02	95.75	86.15	84.46	82.06	85.60	86.29	70.09	72.56	73.43
RegMean	90.87	96.21	85.90	80.06	82.22	86.80	86.72	69.14	70.12	74.19
DOP	89.51	95.71	86.09	80.44	81.21	85.80	86.02	66.54	67.68	73.18
S-Average	90.96	95.88	86.73	80.21	82.19	86.40	86.13	**70.17**	69.51	74.69
S-Task Arithmetic	40.87	50.42	67.55	0.08	28.46	54.20	57.67	54.38	1.22	0.00
S-TIES	91.55	96.34	86.54	86.35	82.83	86.60	**87.27**	68.59	**79.27**	**77.19**
S-KnOTS-TIES	86.77	93.64	83.73	42.61	74.85	80.80	82.32	65.51	52.44	64.66
MOSAIC (Ours)	**91.81**	**96.51**	**87.22**	**86.73**	**83.91**	**87.60**	86.62	**70.17**	78.05	76.19
Order 2										
OPCM	90.53	95.75	86.48	82.18	82.15	86.80	85.69	69.85	70.12	74.94
RegMean	90.87	96.21	85.93	80.21	82.25	86.80	86.67	69.22	70.73	74.44
DOP	90.36	95.83	87.06	78.09	81.20	86.40	85.69	65.27	68.29	72.43
S-Average	90.87	95.79	86.76	80.89	82.24	86.20	85.91	69.77	70.12	74.69
S-Task Arithmetic	14.68	17.80	53.55	0.08	24.96	27.20	51.63	49.72	1.22	0.00
S-TIES	91.30	96.09	86.48	**86.50**	82.55	87.40	**86.78**	68.35	**76.83**	**77.19**
S-KnOTS-TIES	86.86	92.93	83.55	9.63	72.35	79.20	80.52	64.88	45.73	45.36
MOSAIC (Ours)	**91.64**	**96.51**	**87.25**	86.13	**83.93**	**87.60**	86.56	**70.17**	76.22	76.44
Order 3										
OPCM	89.59	95.66	86.57	84.23	81.97	86.00	86.02	69.69	71.95	74.19
RegMean	90.96	96.13	85.96	80.21	82.23	86.80	86.72	69.14	70.73	74.19
DOP	90.19	95.41	86.39	80.97	81.57	86.40	85.64	66.85	67.07	71.18
S-Average	90.96	95.92	86.64	80.82	82.32	86.40	86.07	70.09	69.51	74.44
S-Task Arithmetic	23.63	26.98	58.29	0.30	25.51	34.40	51.47	51.22	0.61	0.00
S-TIES	91.04	96.30	86.51	86.66	83.05	86.20	**87.16**	68.82	76.83	**76.94**
S-KnOTS-TIES	86.52	93.77	83.98	37.38	75.76	82.20	81.94	66.06	54.27	64.66
MOSAIC (Ours)	**91.72**	**96.51**	**87.25**	**87.11**	**83.87**	**87.60**	86.67	**70.17**	**77.44**	75.94
Order 4										
OPCM	90.61	95.88	85.99	84.08	81.29	85.60	85.80	68.90	73.78	75.19
RegMean	90.96	96.21	85.96	79.68	82.28	86.80	**86.83**	69.06	70.73	74.19
DOP	89.33	95.45	86.76	79.83	80.52	86.60	85.47	66.46	65.85	71.68
S-Average	90.78	95.96	86.82	80.59	82.28	86.60	85.91	70.09	70.12	74.44
S-Task Arithmetic	32.34	36.87	59.27	0.08	26.27	36.80	55.28	51.14	2.44	0.00
S-TIES	91.47	**96.55**	86.27	86.43	81.96	85.40	86.72	67.96	77.44	**76.94**
S-KnOTS-TIES	86.60	93.86	84.16	32.30	73.77	80.00	82.10	65.35	53.66	65.16
MOSAIC (Ours)	**91.64**	96.51	**87.28**	**86.66**	**83.91**	**87.60**	86.56	**70.24**	**78.05**	76.19
Order 5										
OPCM	90.70	95.75	86.51	83.32	82.07	86.20	86.02	69.22	70.73	74.94
RegMean	90.96	96.21	86.02	79.91	82.27	86.80	86.67	69.22	70.73	73.93
DOP	90.36	95.96	86.70	79.91	81.28	86.60	84.71	66.85	71.34	71.93
S-Average	90.96	95.92	86.85	80.74	82.22	86.60	86.02	70.09	70.12	74.19
S-Task Arithmetic	41.72	50.46	63.79	0.23	26.77	46.00	53.21	52.17	1.22	0.00
S-TIES	90.78	96.42	86.61	**87.26**	82.67	87.00	**86.72**	68.82	**78.66**	**77.44**
S-KnOTS-TIES	85.84	93.35	84.83	12.36	71.59	81.00	79.38	65.04	42.07	52.88
MOSAIC (Ours)	**91.64**	**96.51**	**87.28**	86.28	**83.92**	**87.60**	86.62	**70.17**	76.22	76.19