ADAPTATION IS NOT RETENTION: REFRAMING CONTINUAL LOW-RANK ADAPTATION THROUGH RESPONSE COORDINATES

Anonymous authors
Paper under double-blind review

ABSTRACT

Low-rank adaptation (LoRA) enables parameter-efficient continual learning, but good current-task fit does not ensure subsequent retention. We identify a controlled *fit–inheritance mismatch*: changing only the task-start down-projection, with the up-projection initialized at zero, preserves the initial model function, yet can yield similar acquisition and substantially different later retention. We study this phenomenon through a response-coordinate view of LoRA, with the down-projection as a *reader* and the up-projection as a *writer*. Under a fixed local response model, we formalize a *response-coordinate bottleneck* in factorized LoRA states: the reader determines the task-response space accessible to adaptation and the local protection boundary carried into subsequent learning, while the writer can only form responses within the reader-exposed coordinates and is optimized in the resulting reader-conditioned geometry. This response-coordinate analysis yields two design principles: *response-supported reader selection* and *reader-conditioned writer optimization*. Building on these principles, we develop **InheritLoRA**, a replay-free continual adaptation method that constructs task-informed readers from response-supported coordinates, optimizes their writers in reader-conditioned geometry, and retains completed low-rank blocks. We evaluate InheritLoRA on T5-large and LLaMA 3.1–8B across the Standard and Long-Sequence continual-learning benchmarks. On the 15-task Long-Sequence benchmark, InheritLoRA matches the leading replay-free method on T5-large with substantially higher adaptation efficiency; on LLaMA 3.1–8B, it surpasses the strongest previously reported replay-free LoRA-based method and even its replay-assisted variant at markedly lower adaptation cost. Controlled interventions further probe the inheritance mechanism, showing that unprotected update components contribute to old-task interference and that inherited reader protection can alter subsequent interference even with the learned function held fixed.

1 INTRODUCTION

Low-rank adaptation (LoRA) adapts large language models by learning low-rank weight updates with a small number of trainable parameters (Hu et al., 2022; Dettmers et al., 2023; Liu et al., 2024). These updates support sequential task adaptation with a frozen pretrained backbone (Wang et al., 2023). In continual learning, each task leaves behind a low-rank factor state that is inherited by subsequent training.

A puzzling observation. A controlled intervention reveals that current-task fit alone does not determine how well a LoRA state is inherited. We vary only the task-start down-projection A_0, while setting the up-projection to $B_0 = 0$ and keeping the task order, training budget, loss, and subsequent update rules unchanged (§ 3.1). Because $B_0 = 0$, these initializations represent the same task-start function, yet they lead to relatively modest changes in task acquisition and substantially different retention after later tasks. We refer to this phenomenon as a *fit–inheritance mismatch*: LoRA states that fit the current task similarly can differ markedly in how they shape subsequent continual learning. *Which properties of the factorized state affect retention during subsequent training?*

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.

A response-coordinate view of LoRA. To study this discrepancy, we view LoRA through a response-coordinate factorization. For an input activation matrix X, a low-rank update $\Delta W = BA$ induces the response

$$\Delta Y = B(AX^\top), \tag{1}$$

where the down-projection A acts as a *reader* that exposes low-dimensional response coordinates, and the up-projection B acts as a *writer* that combines those coordinates into output changes (Zhu et al., 2024). Current-task performance depends on their combined effect, BA, but continual training inherits a particular factorized reader–writer state. Thus, functionally similar low-rank updates need not expose the same coordinates to subsequent optimization.

Why good task fit does not ensure retention. This view exposes a *response-coordinate bottleneck* in the factorized LoRA state. Under a fixed local response model, the reader determines the task-response space accessible to adaptation: even an optimal writer cannot recover task-response directions that the reader fails to expose. When carried into subsequent learning, the reader also defines a local protection boundary, so a deficient reader can leave first-order loss-increasing directions outside the protected subspace. The reader coordinates further determine the geometry in which writer updates act: the same writer-space gradient can induce different response changes under different readers. These properties lead to two principles for continual low-rank adaptation: *response-supported reader selection* and *reader-conditioned writer optimization*.

From response-coordinate principles to method. We develop **InheritLoRA**, a replay-free continual adaptation method that constructs task-informed readers from response-supported coordinates, optimizes their writers in reader-conditioned geometry, and retains completed low-rank blocks. To coordinate successive tasks, each reader proposal is filtered against retained historical reader directions before installation. Experiments on T5-large and LLaMA 3.1–8B show strong continual-learning performance with favorable adaptation efficiency, while controlled interventions further connect unprotected update components and inherited reader protection to subsequent interference.

Our contributions are threefold: **(1) A controlled fit–inheritance mismatch.** We isolate the inheritance effect of LoRA's factorized state by varying only the task-start reader while preserving the initial model function. Despite similar current-task acquisition, different readers can lead to substantially different subsequent retention. **(2) A response-coordinate theory of LoRA inheritance.** Under a fixed local response model, we formalize a *response-coordinate bottleneck*: the reader determines the task-response space accessible to adaptation and the local protection boundary carried into subsequent learning, while also inducing the geometry governing writer optimization. This analysis yields *response-supported reader selection* and *reader-conditioned writer optimization*. **(3) A replay-free continual adaptation method.** We instantiate these principles in **InheritLoRA**. On the 15-task Long-Sequence benchmark, it matches the leading replay-free method on T5-large with substantially higher adaptation efficiency and surpasses the strongest previously reported replay-free LoRA-based method on LLaMA 3.1–8B, as well as its replay-assisted variant, at markedly lower adaptation cost. Controlled interventions on realized updates and inherited reader protection further probe the proposed inheritance mechanism.

2 Related Work

Historical state management. Replay-free continual adaptation typically preserves prior knowledge through model-side state management, including subspace projection, low-rank allocation, adapter aggregation, and task-specific routing. GPM protects historical gradient subspaces, while O-LoRA and InfLoRA extend related projection-based ideas to low-rank adaptation; GORP and ELLA further constrain future updates using projected gradients or accumulated adapter statistics (Saha et al., 2021; Wang et al., 2023; Liang & Li, 2024; Wang et al., 2025a; Das Biswas et al., 2026). Other approaches manage capacity explicitly, for example by allocating low-rank components across tasks or merging successive adapters into a compact retained state (Wang et al., 2025b; Qiao & Mahdavi, 2026). These methods primarily protect, allocate, or consolidate historical information. We study how task-start reader selection and reader-conditioned writer optimization form factorized LoRA states for better inheritance by subsequent continual optimization.

Gradient-informed coordinates. LoRA-GA, LoRA-One, and GoRA use task gradients to initialize low-rank factors, with GoRA additionally using gradient information for rank allocation (Wang

et al., 2024; Zhang et al., 2025; He et al., 2025). These works demonstrate that task signal can be used to identify useful low-rank coordinates at initialization. To study subsequent inheritance in continual learning, we characterize task-start *reader* coordinates using the matched local task response $R^{\text{loc}} = -\eta G X^\top$. This response couples gradient and activation information in output-response space. Under the fixed local model, this view characterizes the reader-exposed subspace that minimizes response-fitting error and connects activation-, gradient-, and joint-response-based constructions within a common criterion.

Factor-coordinate optimization. LoRA's two factors can play asymmetric roles: a fixed or randomly initialized input-side factor can remain effective when the output-side factor is optimized (Zhu et al., 2024). LoRA-Pro and AltLoRA improve factor-gradient optimization, while LoRA-RITE targets optimization invariance under equivalent reparameterizations of the same low-rank map (Wang et al., 2025c; Yu et al., 2025; Yen et al., 2025). Related evidence further shows that similar downstream performance can coexist with different low-rank spectral structure and sequential behavior (Shuttleworth et al., 2025). These works motivate attention to factor geometry, but primarily concern within-task optimization or equivalent coordinate changes. We instead study inheritance across non-equivalent reader states, where the reader determines both the response coordinates exposed to the writer and the geometry of subsequent writer optimization.

3 The Reader Bottleneck in Continual LoRA

3.1 A Controlled Fit–Inheritance Mismatch

To isolate task-start reader effects, we vary only down-projection initialization A_0: a scale-matched random reader or GradResp, constructed from current-task gradients and activations (§ 4.1). All arms use $B_0 = 0$ and share task order, loss, training budget, evaluation protocol, and subsequent update rules. The new block therefore preserves the task-start model function, while its first factor update depends on A_0 (Appendix Proposition A.1). LC measures acquisition; FinalAcc and AvgForget measure subsequent retention (Appendix C.3).

Table 1: Reader-only intervention on T5-large.

Method	Δ LC	Δ Final ACC	Δ Avg. forget.
O-LoRA (Wang et al., 2023)	+0.3	+4.0	−4.0
GORP (Wang et al., 2025a)	+2.2	+6.3	−5.3
IncLoRA	−2.3	+0.4	−1.7

Changes are GradResp minus Random, in percentage points on the 15-task O4 sequence, with one epoch per task, averaged over seeds 42–44, with only A_0 varied. Absolute results are in Appendix Tab. 16. IncLoRA trains a new LoRA adapter for each task, freezes earlier adapters, and uses no orthogonality or norm penalties. Negative changes in AvgForget indicate reduced forgetting and improved retention.

Tab. 1 shows that changing only A_0 can substantially alter subsequent retention. For O-LoRA and GORP, the GradResp intervention changes LC by only $+0.3$ and $+2.2$ points, while increasing FinalAcc by $+4.0$ and $+6.3$ points and reducing AvgForget by 4.0 and 5.3 points, respectively. IncLoRA provides a complementary case: LC decreases by 2.3 points while FinalAcc still increases slightly by 0.4 points; the lower acquisition level also lowers the reference used by AvgForget. Thus, task-start function equivalence does not imply inheritance equivalence.

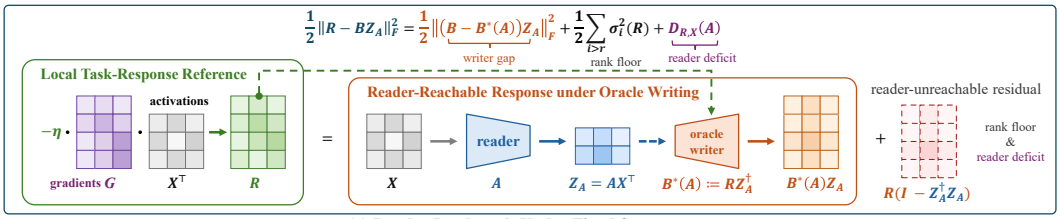
3.2 Reader Bottleneck and Response-Supported Coordinates

At a fixed task state, the effect of a LoRA update depends on the response it induces on task activations. For matched activations $X_t \in \mathbb{R}^{n_t \times d}$,

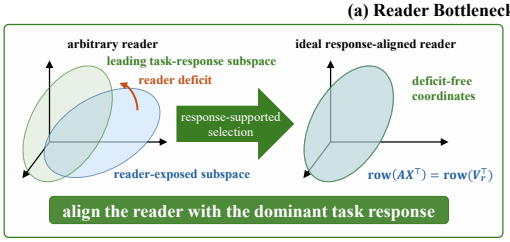
$$\Delta Y_t = \Delta W_t X_t^\top = B_t\big(A_t X_t^\top\big). \tag{2}$$

Thus, the reader exposes response coordinates $A_t X_t^\top$, while the writer combines them into output changes. To compare readers at the same model state, let $G_t = \nabla_{W_t} L_t(W_t)$ and consider the virtual dense step $\delta W_t^{\text{loc}} = -\eta_{\text{loc}} G_t$. On the matched activations, it induces the local task response

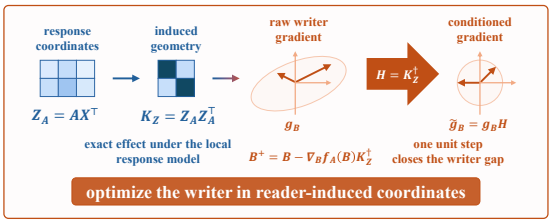
$$R_t^{\text{loc}} := -\eta_{\text{loc}} G_t X_t^\top. \tag{3}$$



(a) Reader Bottleneck Under Fixed State



(b) Principle 1: Response-Supported Reader Selection



(c) Principle 2: Reader-Conditioned Writer Optimization

Figure 1: **Reader bottleneck and design principles. (a)** At a fixed model state and probe, response-fitting error decomposes into the rank floor, reader deficit, and writer gap; oracle writing leaves the rank floor and reader deficit. **(b)** Under the spectral-gap condition of Corollary 3.2, an ideal response-aligned reader exposes the leading task-response subspace and achieves zero reader deficit. **(c)** For fixed (R, X, A), a unit Gram-preconditioned step with $g_B = \nabla_B f_A(B)$ eliminates the writer gap of the local least-squares objective.

At a fixed model state, this response provides an analytical reference for comparing low-rank expressivity across readers.

For integer width $r \geq 1$, let $A \in \mathbb{R}^{r\times d}$ and $B \in \mathbb{R}^{m\times r}$, and write $R := R_t^{\mathrm{loc}}$, $X := X_t$, and $Z_A := AX^\top$. The minimum-norm oracle writer $B^\star(A) := RZ_A^\dagger$ achieves the best response fit within the coordinates exposed by A. Define the *reader deficit* as the excess response-fitting error over the optimal rank-r approximation:

$$\mathcal{D}_{R,X}(A) := \min_{\bar{B}} \frac{1}{2}\|R - \bar{B}Z_A\|_F^2 - \frac{1}{2}\sum_{i>r}\sigma_i^2(R). \tag{4}$$

Singular values are ordered nonincreasingly and are zero beyond the response rank; the proof follows from standard projection and low-rank approximation results and is given in Appendix A.3.

Theorem 3.1 (Reader bottleneck decomposition)**.** *Under the fixed-state setting above, for every reader A and writer B, $\mathcal{D}_{R,X}(A) \geq 0$ and*

$$\frac{1}{2}\|R - BZ_A\|_F^2 = \underbrace{\frac{1}{2}\sum_{i>r}\sigma_i^2(R)}_{\textit{rank floor}} + \underbrace{\mathcal{D}_{R,X}(A)}_{\textit{reader deficit}} + \underbrace{\frac{1}{2}\|(B - B^\star(A))Z_A\|_F^2}_{\textit{writer gap}}. \tag{5}$$

Fig. 1(a) decomposes response-fitting error into the irreducible rank floor, reader deficit, and writer gap. For a fixed reader, even oracle writing eliminates only the writer gap; the remaining reader deficit reflects the restricted response subspace attainable by a rank-r LoRA state.

Corollary 3.2 (Response-optimal reader coordinates)**.** *Under the fixed-state setting of Theorem 3.1, assume $\sigma_r(R) > \sigma_{r+1}(R)$, and let $V_r \in \mathbb{R}^{n_t\times r}$ contain the leading r orthonormal right singular vectors of R. Then*

$$\mathcal{D}_{R,X}(A) = 0 \quad \Longleftrightarrow \quad \mathrm{row}(AX^\top) = \mathrm{row}(V_r^\top). \tag{6}$$

Thus, deficit-free reading exposes the leading task-response subspace (Fig. 1(b)).

Principle 1: Response-supported reader selection. The task-start reader should expose the leading coordinates of the current task response.

Realization and special cases. Appendix A.4 gives the minimum-norm realization and rank-deficient cases. Activation-SVD and Gradient-SVD arise as special cases under isotropy conditions, while GradResp provides the joint gradient–activation construction used in § 4.1.

Local implication for continual protection. In continual learning, a retained reader may also define a protected input subspace. Let $P_A := A^\dagger A$ denote the orthogonal projector onto $\mathrm{row}(A)$. A later weight increment ΔW preserves the layer's action on this subspace when $\Delta W P_A = 0$.

Corollary 3.3 (Unprotected first-order task sensitivity)**.** *Under the fixed-state setting of Theorem 3.1, if $\mathcal{D}_{R,X}(A) > 0$, there exists a rank-one direction $\Delta W \in \mathbb{R}^{m\times d}$ satisfying*

$$\Delta W P_A = 0, \qquad \langle G_t, \Delta W \rangle_F > 0. \tag{7}$$

At a fixed model state, a positive reader deficit implies the existence of first-order loss-increasing directions outside the reader-protected subspace (Appendix A.5). This local existence result does not imply that a larger deficit leads to greater long-horizon forgetting. § 5.2 examines this sensitivity through interventions on realized updates and inherited protection, and measures the resulting changes in old-task loss and accuracy.

3.3 Reader-Conditioned Writer Optimization

Fix a reader A and write $Z_A := AX^\top$ for its exposed response coordinates. To account for their geometry in writer optimization (Tong et al., 2021; Yen et al., 2025), define the Gram matrix and local response-fitting objective

$$K_Z := Z_A Z_A^\top, \qquad f_A(B) := \frac{1}{2}\|R - BZ_A\|_F^2. \tag{8}$$

Corollary 3.4 (Exact Gram-preconditioned writer update)**.** *For fixed R, X, and A,*

$$\nabla_B f_A(B) = (BZ_A - R)Z_A^\top, \tag{9}$$

and the unit-step Gram-preconditioned update

$$B^+ := B - \nabla_B f_A(B) K_Z^\dagger = B + (R - BZ_A)Z_A^\dagger, \tag{10}$$

attains the optimal response fit available to the fixed reader. Appendix A.6 gives the proof.

Thus, under the fixed local least-squares model, Gram preconditioning eliminates the writer gap identified in Theorem 3.1 in a single step (Fig. 1(c)).

Principle 2: Reader-conditioned writer optimization. Writer updates should account for the Gram geometry induced by the coordinates exposed by the selected reader.

4 InheritLoRA: Coordinated Reader–Writer Adaptation

InheritLoRA instantiates the two principles from § 3. Principle 1 defines a response-subspace criterion that admits multiple reader constructions. We use GradResp as the default realization in the main experiments; § 5.2 compares alternative task-informed readers under the same writer protocol. For task t, we construct A_t from current-task data, filter it against historical reader directions, and set $B_t^{(0)} = 0$. We fix A_t and optimize only B_t with reader-conditioned gradient geometry and per-update AdamW-state reset. The base model and historical blocks remain frozen (Fig. 2).

4.1 A GradResp Realization of Response-Supported Reading

At the start of task t, a labeled current-task probe yields the dense-weight loss gradient $G \in \mathbb{R}^{m\times d}$ and matched input activations $X \in \mathbb{R}^{n\times d}$. We take the leading r right singular vectors of $GX^\top$ as the columns of $V_r \in \mathbb{R}^{n\times r}$, so that $V_r^\top$ defines the target response coordinates from Corollary 3.2. We call this joint gradient–activation construction GradResp. To realize these coordinates in input space, we solve a ridge-regularized least-squares problem:

$$\widetilde{A}_t^{(0)} := \arg\min_{A\in\mathbb{R}^{r\times d}} \left\{ \left\| AX^\top - V_r^\top \right\|_F^2 + \lambda \|A\|_F^2 \right\}, \qquad \lambda > 0. \tag{11}$$

The ridge term stabilizes directions weakly constrained by the finite task probe; the closed-form implementation is given in Appendix B.1.

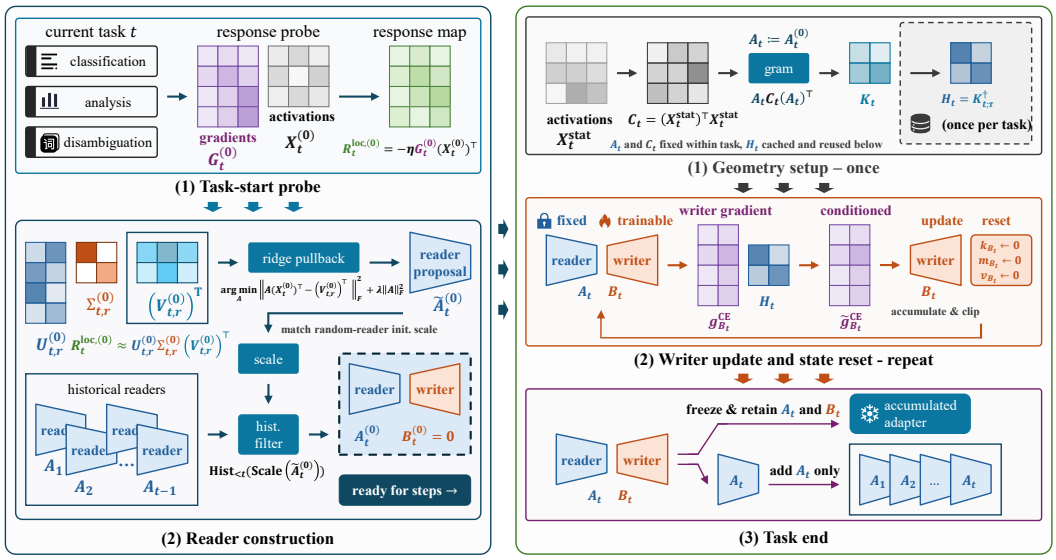


Figure 2: **InheritLoRA with the GradResp reader. (a) Reader initialization: (1)** A current-task probe provides gradients and activations for the response map. **(2)** Ridge pullback, scaling, and historical filtering construct the reader from leading response coordinates; the writer starts at zero. **(b) Writer training: (1)** With the reader fixed, a separate activation sample defines the Gram preconditioner, computed once per task and cached. **(2)** Preconditioned cross-entropy gradients are accumulated and clipped for AdamW updates, with optimizer states reset after each update. **(3)** At task end, the reader–writer block is frozen and retained, and the reader is added to the historical set.

We rescale $\widetilde{A}_t^{(0)}$ using its full-matrix sample standard deviation to match the random-reader initialization scale. To reduce overlap with the row spaces of retained historical readers, we filter the rescaled proposal (Saha et al., 2021; Wang et al., 2023; Liang & Li, 2024) using

$$\mathrm{Hist}_{<t}(A) := \mathcal{R}_A \left[A \left(I_d - \widehat{P}_{<t}^{\mathrm{QR}} \right) \right], \tag{12}$$

where $\widehat{P}_{<t}^{\mathrm{QR}}$ is the reduced-QR projector constructed from the stacked historical readers and $\mathcal{R}_A$ restores the input row norms. This filtering may also remove current response-supported components. Finally, we install

$$A_t^{(0)} = \mathrm{Hist}_{<t}\left(\mathrm{Scale}\left(\widetilde{A}_t^{(0)} \right) \right), \qquad B_t^{(0)} = 0, \tag{13}$$

and keep the reader fixed throughout the task (Fig. 2(a)). Appendix B provides the probing, scaling, and filtering details.

4.2 Reader-Conditioned Writer Optimization

Corollary 3.4 gives an exact inverse-Gram correction for the fixed local response objective. We use this geometry to condition the writer's task-loss gradient during actual cross-entropy training. After installing the task-start reader, we collect a separate current-task activation sample X_t^{stat} from the continual model and construct

$$C_t = (X_t^{\mathrm{stat}})^\top X_t^{\mathrm{stat}}, \qquad K_t = A_t C_t A_t^\top, \qquad H_t = K_{t;\tau}^\dagger, \tag{14}$$

where $K_{t;\tau}^\dagger$ is a truncated spectral pseudoinverse. With A_t and the sampled moment C_t fixed within each task, we compute and cache H_t once (Fig. 2(b.1)). At each training step, we precondition the task-cross-entropy gradient as

$$\widetilde{g}_{B_t}^{\mathrm{CE}} = g_{B_t}^{\mathrm{CE}} H_t. \tag{15}$$

We accumulate and clip these preconditioned gradients, then update B_t with AdamW to minimize the current task's cross-entropy loss.

Table 2: Replay-free continual accuracy and first-task efficiency. **Best**/second-best.

Methods	Standard CL Benchmark (SC)				Long Sequence Benchmark (LS)				Efficiency Metrics		
	O1	O2	O3	Avg.	O4	O5	O6	Avg.	Trainable params. (M)	Train (ms/step)	Infer (tok/s)
T5-Large											
SeqLoRA* (Wang et al., 2023)	44.6	32.7	53.7	43.7	2.3	0.6	1.9	1.6	2.359	180.6	284.5
IncLoRA* (Wang et al., 2023)	66.0	64.9	68.3	66.4	63.3	58.5	61.7	61.2	2.359	**167.6**	301.7
O-LoRA (Wang et al., 2023)	75.4	75.7	76.3	75.8	72.3	64.8	71.6	69.6	2.359	179.8	306.0
SPARTA (Liao et al., 2026)	73.7	70.5	73.8	72.7	71.5	70.5	68.0	70.0	**11.403**	**468.7**	**147.5**
CLoRA (Lu et al., 2025)	79.7	79.1	78.2	79.0	73.6	66.4	72.4	70.8	2.359	231.2	325.4
CLoRA + O-LoRA (Lu et al., 2025)	76.6	75.5	75.4	75.8	72.0	67.3	77.3	72.2	2.359	243.2	319.3
ASO-LoRA (Wang et al., 2026)	76.2	77.5	77.3	77.0	73.7	67.2	**77.4**	72.8	2.359	190.3	292.9
ELLA (Das Biswas et al., 2026)	**80.0**	**80.0**	**79.8**	**79.9**	73.4	72.0	75.4	73.6	2.359	173.2	303.0
GORP (Wang et al., 2025a)	79.7	79.9	79.7	79.8	76.1	76.2	75.6	**76.0**	**51.905**	**521.5**	**364.2**
InheritLoRA (ours)	78.4	78.7	78.1	78.4	**76.4**	**76.4**	74.9	75.9	**1.180**	174.0	314.4
LLaMA 3.1–8B											
O-LoRA* (Liao et al., 2026)	71.8	72.2	72.8	72.3	73.1	69.4	71.6	71.4	3.408	577.9	**321.3**
ELLA (Das Biswas et al., 2026)	77.8	77.2	77.7	77.6	72.9	72.8	76.8	74.2	3.408	583.3	305.2
SeqLoRA* (Liao et al., 2026)	79.9	79.0	80.0	79.6	74.2	73.7	76.5	74.8	3.408	**547.8**	318.4
SPARTA (Liao et al., 2026)	**81.4**	**80.7**	**80.5**	**80.9**	80.7	77.7	81.5	80.0	**27.787**	**1724.7**	**136.1**
InheritLoRA (ours)	79.4	79.9	80.4	79.9	**82.7**	**83.2**	**81.9**	**82.6**	**1.311**	653.4	304.3

Baseline accuracies (%) come from cited papers; asterisks mark third-party results. Local reimplementations are reported in § C.5; **under our local settings, GORP's reproduced accuracies are substantially lower than its source-reported results.** Efficiency measurement protocols and implementation details are in § C.7.

Per-update optimizer-state reset. After each writer update, we clear AdamW's step counter and both moment buffers while retaining the updated weights and global learning-rate schedule (Fig. 2(b.2)). With these optimizer settings, each update's AdamW moments depend only on the current processed gradient. § 5.3 evaluates reset separately from Gram preconditioning; Appendix B.3 gives the equivalent explicit update and verifies the implementation. At task end, we freeze and retain (A_t, B_t), adding A_t to the historical reader set (Fig. 2(b.3)).

5 Experiments

Experimental Setup. We evaluate T5-large and LLaMA 3.1–8B on four-task Standard CL (SC) and 15-task Long-Sequence (LS). SC comprises DBpedia, Amazon, Yahoo, and AG News in orders O1–O3; LS uses 15 tasks in orders O4–O6. Final average accuracy (FinalAcc) averages task accuracies at the end of a sequence; benchmark means weight the three orders equally. Our runs use seeds 42–44, rank 8, and one epoch per task; matched reader and writer controls use O4 with the same seeds. InheritLoRA uses only current-task examples and retains frozen low-rank task blocks, without historical-example replay or oracle task identifiers. We also report learning accuracy (LC), average forgetting (AvgForget), and backward transfer (BWT). Appendix C.3 lists task sequences and metric definitions; implementation details appear in Appendices B and C.

5.1 Continual Learning Performance

Main results. Tab. 2 compares replay-free methods on the same benchmarks and task orders, retaining source training protocols (Appendix C.4). On T5-large LS, InheritLoRA reaches 75.9% FinalAcc, above the strongest listed frozen-backbone baseline, ELLA (73.6%), and comparable to GORP (76.0%), which also updates full-rank backbone matrices. On LLaMA 3.1–8B LS, it reaches 82.6%, exceeding the strongest listed replay-free baseline, SPARTA (80.0%). On SC, it remains below ELLA on T5 and SPARTA on LLaMA; gains therefore concentrate on LS. Across O4–O6, BWT and AvgForget average -2.52 and 2.81 points for T5-large, and -0.29 and 0.84 points for LLaMA 3.1–8B, respectively (Appendix D.1). Replay-assisted comparisons also retain source protocols (Tab. 3): InheritLoRA exceeds SPARTA+Replay's 74.3%/82.2% LS scores on T5/LLaMA, while Recurrent-KIF reaches a higher 77.8% on T5 with historical-example replay and a training budget of ten epochs per task.

Parameter and computational efficiency. In matched first-task measurements, InheritLoRA uses 50.0% and 61.5% fewer trainable parameters than O-LoRA on T5 and LLaMA, respectively. Training-step cost and inference throughput are similar on T5; on LLaMA, steps take 13.1% longer

Table 3: Replay-assisted comparisons and first-task efficiency. Best in **bold**.

Methods	Replay	Standard CL Benchmark (SC)				Long Sequence Benchmark (LS)				Efficiency Metrics		
		O1	O2	O3	Avg.	O4	O5	O6	Avg.	Trainable params. (M)	Train (ms/step)	Infer (tok/s)
T5-Large												
SPARTA + Replay (Liao et al., 2026)	Yes	77.0	75.6	75.2	75.9	75.6	73.2	74.1	74.3	**11.403**	**442.6**	**152.5**
Recurrent-KIF† (Feng et al., 2025)	Yes	–	–	–	**78.4**	–	–	–	**77.8**	2.359	**169.5**	298.4
InheritLoRA (ours)	No	**78.4**	**78.7**	**78.1**	**78.4**	**76.4**	**76.4**	**74.9**	75.9	**1.180**	174.0	**314.4**
LLaMA 3.1–8B												
SPARTA + Replay (Liao et al., 2026)	Yes	**80.6**	**81.0**	**80.7**	**80.8**	**83.1**	81.7	81.8	82.2	**27.787**	**1749.9**	**131.8**
InheritLoRA (ours)	No	79.4	79.9	80.4	79.9	82.7	**83.2**	**81.9**	**82.6**	**1.311**	**653.4**	**304.3**

Accuracy conventions and our results follow Tab. 2. †Recurrent-KIF: order-averaged OP, replay, 10 epochs/task; dashes: unreported (sources: citations, Appendix C.4).

and throughput is 5.3% lower (Tab. 2). Trainable writer size is fixed per task, while retained state grows linearly with task count. Appendices C.7 and D.11 detail task-start initialization costs, the efficiency measurement protocol, and the state retained across tasks.

5.2 Reader Selection and Inheritance Diagnostics

Reader selection. On T5-large/O4, Activation-SVD, Gradient-SVD, and GradResp raise FinalAcc from the scale-matched Random reader's 66.80% to 76.16–76.89%, with mean LC near 78% (Tab. 4). Gradient-SVD has the highest mean FinalAcc. On LLaMA 3.1–8B/O4, GradResp raises FinalAcc from 78.82% to 83.03% and lowers AvgForget from 2.46 to 0.47 points relative to Random (Appendix D.3.2). Gradient-SVD and Activation-SVD attain the response-subspace optimum of Corollary 3.2 under the respective isotropy conditions in Appendix A.4. We choose GradResp for its simple, direct implementation of response-supported reader selection (§ 4.1).

Table 4: Reader constructions under response-coordinate selection. **Best**/second-best.

Reader	LC (%)	FinalAcc (%)	BWT (pp)
Scale-matched random	**78.31 ± 0.31**	66.80 ± 1.22	−12.34 ± 1.08
Activation-SVD	77.54 ± 0.39	76.16 ± 0.12	−1.48 ± 0.30
Gradient-SVD	77.94 ± 0.49	**76.89 ± 0.36**	**−1.12 ± 0.69**
GradResp	78.14 ± 0.68	76.39 ± 0.75	−1.88 ± 0.70

T5-large/O4; mean ± sample SD. Readers are fixed; writers use Gram preconditioning and per-update reset. GradResp reuses T5 Full (Tab. 5).

From local vulnerability to realized interference. We test whether unprotected sensitivity directions contribute to interference from realized updates. Alongside accuracy, we measure the negative log-likelihood (NLL) of the correct target tokens. We average over valid tokens within each example and then over examples; lower NLL indicates better predictive fit. Deleting realized update components along the unprotected sensitivity directions of Corollary 3.3 improves old-task loss beyond all 20 matched random deletions and norm-matched shrinkage in $20/24$ transition–seed cases. Full deletion narrows the absolute Random–GradResp old-task damage gap in $11/12$ matched pairs, linking these components to reader-dependent interference. New-task loss worsens in $19/24$ cases, with greater full-deletion cost for Random in all 12 pairs (Appendix D.7).

Inherited protection and interference. At a fixed learned function, we swap only the retained reader used for future projection across three task transitions (Appendix D.8). Relative to Random protection, GradResp protection lowers complete-test old-task NLL in all six CB→WiC parent–seed comparisons. In all six RTE→IMDB comparisons it lowers first-step gradient–update inner products and old-task NLL increments, but yields no mean complete-test NLL benefit at task end. For DBpedia→AG News, old-task NLL decreases on the heldout sample in all six comparisons but increases on the complete test set in five, with little change in old-task accuracy. Inherited protection affects subsequent interference; task-end retention benefits vary with the task transition and the sample used to evaluate old-task retention.

Task-end deficit and later forgetting. We examine whether task-end reader deficit consistently ranks later forgetting, using the mean of the energy-normalized deficits across adapted layers. Across 168 task-end states, deficit–damage correlations vary with adjustment for task, seed, acqui-

sition accuracy, and acquisition diagnostic NLL (Appendix D.6). Rank- and spectrum-preserving interventions separate this measure by 1.40–5.65 pp in all nine window–seed pairs. Immediately after learning the old task, branch accuracies differ by at most 1.79 pp, while NLL is less closely matched. Higher-deficit branches have larger old-task accuracy drops after the next task in four pairs, smaller drops in four, and one tie. At the RTE stream endpoint, the higher-deficit branch has a larger signed accuracy drop for seed 43 and a smaller drop for seed 44 (Appendix D.9). Together, these diagnostics show that inherited protection affects subsequent interference, while the measured task-end deficit does not consistently rank later forgetting.

5.3 Writer Optimization and Continual Retention

Similar acquisition, different retention. On T5-large/O4 at 10^{-3}, Full and the variant without either component attain 78.14% and 78.07% LC, yet 76.39% and 72.56% FinalAcc, with AvgForget of 2.38 and 6.27 points (Tab. 5). Removing preconditioning or reset alone also lowers mean FinalAcc and increases mean forgetting. On LLaMA at 10^{-4}, Full has the highest mean FinalAcc (83.03%) and lowest mean AvgForget (0.47 points).

Table 5: Acquisition and retention under writer component ablations. **Best**/second-best.

Configuration	T5-large (10^{-3})			LLaMA 3.1–8B (10^{-4})		
	LC (%)	FinalAcc (%)	AvgForget (pp)	LC (%)	FinalAcc (%)	AvgForget (pp)
Full	**78.14 ± 0.68**	**76.39 ± 0.75**	**2.38 ± 0.47**	82.75 ± 0.26	**83.03 ± 0.08**	**0.47 ± 0.05**
w/o precond.	77.77 ± 0.24	75.44 ± 1.39	3.13 ± 1.15	82.43 ± 0.95	82.17 ± 1.14	0.88 ± 0.27
w/o reset	77.98 ± 0.71	75.21 ± 1.26	3.30 ± 0.86	82.61 ± 0.44	82.74 ± 0.44	0.64 ± 0.42
w/o both	78.07 ± 0.63	72.56 ± 1.27	6.27 ± 2.03	**83.51 ± 0.05**	82.89 ± 0.45	1.06 ± 0.26

GradResp/O4; mean ± sample SD. Full: Gram preconditioning + per-update reset (Appendix D.4).

Reader-conditioned local fitting. To isolate the geometric effect, we fix the reader, writer, and local least-squares objective at 12 T5-large/O4 checkpoints and give each update direction its optimal scalar step (Appendix D.4.5). Across 1,728 layer–state checks, exact Gram conditioning closes the fixed-reader writer gap to numerical precision; Identity and Diagonal retain mean gap ratios of 0.2711 and 0.3474. Applying the implementation's 10^{-2} spectral cutoff to the same matched-probe Gram matrix leaves Full with a mean gap ratio of 0.01193.

Acquisition–retention profiles across learning rates. With reset enabled, conditioning lowers mean AvgForget at all four tested T5 rates from 10^{-4} to 3×10^{-3} (Appendix D.4.2). At 10^{-4}, lower LC accompanies a 2.13-point FinalAcc reduction. At 3×10^{-4}, Full and NoPrecond reach their highest mean FinalAcc in the tested grid (76.78% and 76.74%), with lower LC and AvgForget for Full. Conditioning improves mean FinalAcc by 0.95 and 3.87 points at 10^{-3} and 3×10^{-3}, respectively. At 10^{-3}, Diagonal also attains mean FinalAcc comparable to Full under CE and AdamW (Appendix D.4.4). At higher rates, LLaMA deteriorates at 10^{-3} and T5 at 10^{-2} (Tab. 29; Appendix D.4.2). Conditioning thus changes the acquisition–retention balance across learning rates. With conditioning enabled, reset improves mean FinalAcc at each of the four rates from 10^{-4} to 3×10^{-3}; frequency controls appear in Appendix D.4.3.

6 Conclusion

Current-task fit alone does not determine how a LoRA state will be inherited. Under a fixed local response model, our response-coordinate analysis identifies a *response-coordinate bottleneck*: the reader determines the task-response space accessible to adaptation and the local protection boundary carried into subsequent learning, while also inducing the geometry governing writer optimization. This analysis yields response-supported reader selection and reader-conditioned writer optimization, which we instantiate in **InheritLoRA**. Controlled interventions further show that unprotected update components contribute to old-task interference and that inherited reader protection can alter subsequent interference even with the learned function held fixed, while the measured reader deficit does not consistently rank later forgetting. Across both backbones, InheritLoRA delivers a strong accuracy–efficiency trade-off without historical-example replay: it remains competitive with the leading replay-free method on T5-large at substantially lower adaptation cost, while on LLaMA 3.1–8B it improves upon the strongest previously reported replay-free LoRA-based method and its replay-assisted variant.

AI USE STATEMENT

Generative AI tools assisted with manuscript writing, revision, and LaTeX preparation. They also helped develop proofs and refine the authors' proposed theoretical claims, and provided feedback on manuscript clarity and consistency. Additional assistance covered research-code implementation and debugging, experimental planning, and result analysis. Checks included numerical consistency tests, implementation tests, and comparisons of reported values with experiment logs and source records. The authors take responsibility for the research decisions, verification of the theoretical and empirical claims, and final scientific content, including all text, proofs, research code, and analyses developed with assistance from these tools.

REPRODUCIBILITY STATEMENT

Appendix A provides the theoretical assumptions and proofs. Appendix B details the algorithm, reader construction, writer conditioning, optimizer-state reset, and current-task data interfaces. Appendix C documents the datasets, task orders, training protocols, evaluation metrics, seed conventions, and baseline provenance. Appendix D reports seed-level results, component ablations, and the protocols and results of the mechanism diagnostics.

REFERENCES

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

Jiefeng Chen, Timothy Nguyen, Dilan Gorur, and Arslan Chaudhry. Is forgetting less a good inductive bias for forward transfer? In *International Conference on Learning Representations (ICLR)*, 2023.

Shristi Das Biswas, Yue Zhang, Anwesan Pal, Radhika Bhargava, and Kaushik Roy. ELLA: Efficient lifelong learning for adapters in large language models. In *Proc. Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pp. 1907–1924, 2026.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2023.

Wenyu Du, Shuang Cheng, Tongxu Luo, Zihan Qiu, Zeyu Huang, Ka Chun Cheung, Reynold Cheng, and Jie Fu. Unlocking continual learning abilities in language models. In *Findings of the Association for Computational Linguistics: EMNLP*, pp. 6503–6522, 2024.

Yujie Feng, Xujia Wang, Zexin Lu, Shenghong Fu, Guangyuan Shi, Yongxin Xu, Yasha Wang, Philip S. Yu, Xu Chu, and Xiao-Ming Wu. Recurrent knowledge identification and fusion for language model continual learning. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 4th edition, 2013.

Jacopo Graldi, Alessandro Breccia, Giulia Lanzillotta, Thomas Hofmann, and Lorenzo Noci. The importance of being lazy: Scaling limits of continual learning. In *Proc. International Conference on Machine Learning (ICML)*, pp. 20326–20367, 2025.

Haonan He, Peng Ye, Yuchen Ren, Yuan Yuan, Luyang Zhou, Shucun Ju, and Lei Chen. GoRA: Gradient-driven adaptive low rank adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

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 (ICLR)*, 2022.

Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23638–23647, 2024.

Huanxuan Liao, Shizhu He, Yupu Hao, Yequan Wang, Wenhao Teng, Xiangwen Liao, Jun Zhao, and Kang Liu. Spectral disentanglement: Rank-aware task adaptation for rehearsal-free continual learning in LLMs. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 7360–7376, 2026.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *Proc. International Conference on Machine Learning (ICML)*, pp. 32100–32121, 2024.

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

Binghang Lu, Zheyuan Deng, Runyu Zhang, Bing Hu, Yunhan Zhao, Yuan Tian, Changhong Mou, Guang Lin, and Xiaomin Li. Muon-OGD: Muon-based spectral orthogonal gradient projection for LLM continual learning. *arXiv preprint arXiv:2605.08949*, 2026.

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 *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

Nikhil Shivakumar Nayak, Krishnateja Killamsetty, Ligong Han, Abhishek Bhandwaldar, Prateek Chanda, Kai Xu, Oleg Silkin, Mustafa Eyceoz, Hao Wang, Aldo Pareja, and Akash Srivastava. Sculpting subspaces: Constrained full fine-tuning in LLMs for continual learning. In *International Conference on Learning Representations (ICLR)*, 2026.

Fuli Qiao and Mehrdad Mahdavi. Learn more, but bother less: Parameter-efficient continual learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024.

Fuli Qiao and Mehrdad Mahdavi. Merge before forget: A single LoRA continual learning via continual merging. In *International Conference on Learning Representations (ICLR)*, 2026.

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

Reece Shuttleworth, Jacob Andreas, Antonio Torralba, and Pratyusha Sharma. LoRA vs full fine-tuning: An illusion of equivalence. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

Tian Tong, Cong Ma, and Yuejie Chi. Accelerating ill-conditioned low-rank matrix estimation via scaled gradient descent. *Journal of Machine Learning Research*, 22(150):1–63, 2021.

Chenxu Wang, Yilin Lyu, Zicheng Sun, and Liping Jing. Continual gradient low-rank projection fine-tuning for LLMs. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 14815–14829, 2025a.

Shaowen Wang, Linxi Yu, and Jian Li. LoRA-GA: Low-rank adaptation with gradient approximation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024.

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

Xiequn Wang, Zhan Zhuang, and Yu Zhang. PLAN: Proactive low-rank allocation for continual learning. In *Proc. IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2909–2918, 2025b.

Yitong Wang, Xue Han, Wenchun Gao, Qian Hu, Jiahui Wang, Ziqing Wang, Lijun Mei, and Junlan Feng. Soft orthogonal low-rank adaptation for knowledge sharing in large language model continual learning. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 18472–18487, 2026.

Zhengbo Wang, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. LoRA-Pro: Are low-rank adapters properly optimized? In *International Conference on Learning Representations (ICLR)*, 2025c.

Shuo Yang, Kun-Peng Ning, Yu-Yang Liu, Jia-Yu Yao, Yong-Hong Tian, Yibing Song, and Li Yuan. Is parameter collision hindering continual learning in LLMs? In *Proc. International Conference on Computational Linguistics (COLING)*, pp. 4243–4259, 2025.

Jui-Nan Yen, Si Si, Zhao Meng, Felix Yu, Venkata Sai Surya Subramanyam Duvvuri, Inderjit S. Dhillon, Cho-Jui Hsieh, and Sanjiv Kumar. LoRA done RITE: Robust invariant transformation equilibration for LoRA optimization. In *International Conference on Learning Representations (ICLR)*, 2025.

Xin Yu, Yujia Wang, Jinghui Chen, and Lingzhou Xue. AltLoRA: Towards better gradient approximation in low-rank adaptation with alternating projections. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

Yuanhe Zhang, Fanghui Liu, and Yudong Chen. LoRA-One: One-step full gradient could suffice for fine-tuning large language models, provably and efficiently. In *Proc. International Conference on Machine Learning (ICML)*, pp. 75513–75574, 2025.

Jiacheng Zhu, Kristjan Greenewald, Kimia Nadjahi, Haitz Sáez De Ocáriz Borde, Rickard Brüel Gabrielsson, Leshem Choshen, Marzyeh Ghassemi, Mikhail Yurochkin, and Justin Solomon. Asymmetry in low-rank adapters of foundation models. In *Proc. International Conference on Machine Learning (ICML)*, pp. 62369–62385, 2024.

A Theoretical Details and Proofs

The proofs follow § 3: reader initialization and the local response model lead to the reader bottleneck, response-optimal coordinates, unprotected task sensitivity, and the exact Gram-preconditioned writer update. Appendix A.7 then gives the complementary coordinate-equivariance result.

A.1 Reader Initialization and the First Product Update

Proposition A.1 (Reader initialization controls the first product update)**.** *For $\mathcal{L}(A, B) = \ell(W + BA)$, let $G = \nabla_M \ell(W + M)|_{M=BA}$. The task-loss gradients are*

$$\nabla_B \mathcal{L} = GA^\top, \qquad \nabla_A \mathcal{L} = B^\top G. \tag{16}$$

Hence, for $B_0 = 0$ and $G_0 = \nabla_M \ell(W + M)|_{M=0}$, $\nabla_{A_0} \mathcal{L} = 0$ and $\nabla_{B_0} \mathcal{L} = G_0 A_0^\top$. Under one simultaneous Euclidean factor step with writer learning rate $\eta_B > 0$,

$$A_1 = A_0, \qquad B_1 = -\eta_B G_0 A_0^\top, \qquad M_1 := B_1 A_1 = -\eta_B G_0 A_0^\top A_0. \tag{17}$$

Moreover,

$$\langle G_0, M_1 \rangle_F = -\eta_B \|G_0 A_0^\top\|_F^2 \le 0, \tag{18}$$

with strict inequality if and only if $G_0 A_0^\top \neq 0$.

Proof of Proposition A.1. For $\mathcal{L}(A, B) = \ell(W + BA)$, let $G = \nabla_M \ell(W + M)|_{M=BA}$. The differential is

$$\mathrm{d}\mathcal{L} = \langle G, (\mathrm{d}B)A + B(\mathrm{d}A) \rangle_F, \tag{19}$$

which yields Equation 16. Substituting $B_0 = 0$ into the simultaneous Euclidean step gives the factors and product in Equation 17. Finally,

$$\begin{aligned} \langle G_0, M_1 \rangle_F &= -\eta_B \operatorname{tr}(G_0^\top G_0 A_0^\top A_0) \\ &= -\eta_B \operatorname{tr}(A_0 G_0^\top G_0 A_0^\top) = -\eta_B \|G_0 A_0^\top\|_F^2. \end{aligned} \tag{20}$$

This quantity is strictly negative exactly when $G_0 A_0^\top \neq 0$. Q.E.D.

First-step optimization and oracle profiling. The Euclidean first step shows how A_0 shapes the initial product update; the oracle writer $R(A_0 X^\top)^\dagger$ measures fixed-reader expressivity. In ordinary joint factor training, $B^\top G$ can update the reader once $B \neq 0$, while regularization and optimizer state can also contribute to changes in the reader parameters.

A.2 Local Task-Response Model

Fix one dense layer, model state, and matched probe. Let $G = \nabla_W \mathcal{L}(W) \in \mathbb{R}^{m \times d}$, and let $X \in \mathbb{R}^{n \times d}$ contain the corresponding activations. The Frobenius-proximal first-order model gives

$$\delta W^{\mathrm{loc}} := \arg\min_{\Delta} \left\{ \langle G, \Delta \rangle_F + \frac{1}{2\eta_{\mathrm{loc}}} \|\Delta\|_F^2 \right\} = -\eta_{\mathrm{loc}} G. \tag{21}$$

Applying this virtual step to the matched activations gives $R^{\mathrm{loc}} = \delta W^{\mathrm{loc}} X^\top = -\eta_{\mathrm{loc}} G X^\top$, as used in Equation 3. This virtual response measures fixed-state expressivity; the implemented writer is trained on task cross-entropy (§ 4.2).

A.3 Proof of the Reader Bottleneck Decomposition

The decomposition in the main theorem specializes a general projection identity. For an integer width $r \ge 1$, arbitrary matched $X \in \mathbb{R}^{n \times d}$, and $R = \Delta X^\top \in \mathbb{R}^{m \times n}$, with $\Delta \in \mathbb{R}^{m \times d}$, set $Z_A = AX^\top$, $P_{Z_A} = Z_A^\dagger Z_A$, and $\mathcal{E}_R(A) = \min_{\bar{B}} \frac{1}{2} \|R - \bar{B} Z_A\|_F^2$, and let $q = \min\{r, \operatorname{rank}(R)\}$. For the proofs, abbreviate the rank floor as

$$\mathcal{E}_r^\star(R) := \frac{1}{2} \sum_{i > r} \sigma_i^2(R) = \frac{1}{2} \sum_{i > q} \sigma_i^2(R), \tag{22}$$

where singular values beyond the response rank are zero. Taking $R = R_t^{\mathrm{loc}} = -\eta_{\mathrm{loc}} G_t X_t^\top$, $X = X_t$, and $\Delta = \delta W_t^{\mathrm{loc}}$ yields the decomposition in Theorem 3.1.

Proof of Theorem 3.1. Set $Z = AX^\top$, $P_Z = Z^\dagger Z$, and $B^\star = RZ^\dagger$. The standard matrix least-squares projection identity gives $B^\star Z = RP_Z$ and

$$R - BZ = R(I - P_Z) + (B^\star - B)Z. \tag{23}$$

The two terms have orthogonal row spaces, and therefore

$$\frac{1}{2}\|R - BZ\|_F^2 = \frac{1}{2}\|R(I - P_Z)\|_F^2 + \frac{1}{2}\|(B - B^\star)Z\|_F^2. \tag{24}$$

Since every fitted response BZ has rank at most r, the Eckart–Young–Mirsky theorem gives

$$\frac{1}{2}\|R(I - P_Z)\|_F^2 \geq \mathcal{E}_r^\star(R). \tag{25}$$

Using the definition $\mathcal{D}_{R,X}(A) = \frac{1}{2}\|R(I - P_Z)\|_F^2 - \mathcal{E}_r^\star(R)$ yields Equation 5 and $\mathcal{D}_{R,X}(A) \geq 0$. For responses $R = \Delta X^\top$, Appendix A.4 establishes attainability of the rank floor and characterizes the reader coordinates that achieve zero deficit. Q.E.D.

A.4 Response-Optimal Coordinates and Minimum-Norm Realization

General rank-deficient response-optimality statement. Let $R = \Delta X^\top = U\Sigma V^\top$, set $p = \mathrm{rank}(R)$, and let $q = \min\{r, p\}$. If $p > 0$, there exists a width-r reader $A^\star$ such that

$$\min_{A \in \mathbb{R}^{r \times d}} \mathcal{E}_R(A) = \frac{1}{2}\sum_{i>q} \sigma_i^2(R), \qquad \mathrm{row}(A^\star X^\top) = \mathrm{row}(V_q^\top). \tag{26}$$

If $p = 0$, every reader attains zero error. When $p \geq r$ and $\sigma_r > \sigma_{r+1}$, zero deficit is characterized by Equation 6; the boundary spectral gap makes that active response subspace unique.

Proof of Corollary 3.2 and the general statement. We prove the general fixed-response form above and derive the zero-deficit characterization under the corollary's spectral-gap condition. The main-text setting follows from $R = R_t^{\mathrm{loc}}$ and $X = X_t$. Let $p = \mathrm{rank}(R)$, $q = \min\{r, p\}$, and write $R = U\Sigma V^\top$. If $p = 0$, then $R = 0$ and every reader attains zero error. Assume henceforth that $p > 0$. Since $R = \Delta X^\top$, $R^\top = X\Delta^\top$, and therefore $\mathrm{col}(V_q) \subseteq \mathrm{col}(X)$. Hence the fixed response coordinates

$$Z^\star = \begin{bmatrix} V_q^\top \\ 0 \end{bmatrix} \tag{27}$$

are feasible, for example through

$$A^\star = Z^\star (X^\top)^\dagger, \tag{28}$$

because the rows of $Z^\star$ lie in $\mathrm{row}(X^\top)$. This proves the optimum and realizability in the general statement; when $p \geq r$, it also yields the feasible response frame used in Equation 6.

To refine the deficit in Equation 4 quantitatively, let $P = P_{Z_A}$, $\lambda_i = \sigma_i^2$, and $\alpha_i = v_i^\top P v_i$. If $r \leq p$, so $q = r$, then

$$2\mathcal{D}_{R,X}(A) = \sum_{i \leq r} \lambda_i(1 - \alpha_i) - \sum_{i>r} \lambda_i \alpha_i. \tag{29}$$

Set

$$\delta := \sum_{i \leq r}(1 - \alpha_i) = \|(I - P)V_r\|_F^2. \tag{30}$$

Since $\mathrm{rank}(P) \leq r$, $\sum_{i>r} \alpha_i \leq \delta$, and therefore

$$2\mathcal{D}_{R,X}(A) \geq (\lambda_r - \lambda_{r+1})\delta. \tag{31}$$

If $r > p$, so $q = p$, then $\mathcal{E}_r^\star(R) = 0$ and

$$2\mathcal{D}_{R,X}(A) = \sum_{i=1}^{p} \lambda_i(1 - \alpha_i) \geq \lambda_p \|(I - P)V_p\|_F^2. \tag{32}$$

With $\sigma_{p+1} := 0$, the two cases give the quantitative bound

$$\mathcal{D}_{R,X}(A) \geq \frac{1}{2}\big(\sigma_q^2 - \sigma_{q+1}^2\big)\|(I - P_{Z_A})V_q\|_F^2. \tag{33}$$

When $p \geq r$ and $\sigma_r > \sigma_{r+1}$, zero deficit and Equation 33 force $(I - P_{Z_A})V_r = 0$. Since $\mathrm{rank}(P_{Z_A}) \leq r$ and $\mathrm{col}(V_r)$ has dimension r, this implies $\mathrm{row}(Z_A) = \mathrm{row}(V_r^\top)$; conversely, this row-space equality directly yields zero reader deficit.

The active response subspace is unique under the stated spectral gap. Its reader realization A admits invertible row-coordinate changes and perturbations N satisfying $NX^\top = 0$, which preserve that subspace. When $p < r$, zero deficit requires only $\mathrm{row}(V_p^\top) \subseteq \mathrm{row}(Z_A)$, so this condition does not uniquely determine the remaining directions in the reader row space. Q.E.D.

Minimum-norm realization. Under the conditions of Corollary 3.2, fix the response frame $AX^\top = V_r^\top$. Among readers satisfying this constraint, the unique minimum-Frobenius-norm realization (denoted by me) is (Golub & Van Loan, 2013)

$$A^{\mathrm{me}} = V_r^\top (X^\top)^\dagger. \tag{34}$$

Proof. The complete solution family of $AX^\top = V_r^\top$ is

$$A = V_r^\top (X^\top)^\dagger + N\big(I_d - X^\top (X^\top)^\dagger\big) = A^{\mathrm{me}} + N(I_d - P_X), \tag{35}$$

where $N \in \mathbb{R}^{r\times d}$ is arbitrary and $P_X = X^\dagger X = X^\top (X^\top)^\dagger$. The Moore–Penrose solution satisfies $A^{\mathrm{me}}(I_d - P_X) = 0$, and hence is Frobenius orthogonal to every extension:

$$\|A^{\mathrm{me}} + N(I_d - P_X)\|_F^2 = \|A^{\mathrm{me}}\|_F^2 + \|N(I_d - P_X)\|_F^2. \tag{36}$$

This proves the solution family, the zero-extension identity, and uniqueness of the minimum-Frobenius representative claimed in Equation 34.

Q.E.D.

Special cases: activation- and gradient-based readers. The response criterion connects Activation-SVD and Gradient-SVD through two sufficient isotropy conditions. Write the compact singular value decomposition as $X = U_X \Sigma_X Q^\top$, so $P_X = QQ^\top$ projects onto the activation span in input space. For $R = -\eta_{\mathrm{loc}} G X^\top$, the leading response coordinates are determined by

$$R^\top R = \eta_{\mathrm{loc}}^2 X G^\top G X^\top. \tag{37}$$

We work under the corollary's positive rank-r spectral-gap condition; it identifies a unique leading response subspace.

Activation-SVD. Suppose the gradient map preserves inner products on the activation span up to a positive scalar γ:

$$P_X G^\top G P_X = \gamma P_X, \qquad \gamma > 0. \tag{38}$$

Since $X = XP_X$, Equation 37 becomes $R^\top R = \eta_{\mathrm{loc}}^2 \gamma X X^\top$. Its leading eigenvectors are therefore the leading left singular vectors $U_{X,r}$ of X. Let Q_r contain the corresponding right singular vectors and $\Sigma_{X,r}$ the positive singular values. The Activation-SVD reader $A_{\mathrm{act}} = Q_r^\top$ exposes $A_{\mathrm{act}} X^\top = \Sigma_{X,r} U_{X,r}^\top$. Its response row space is thus the optimum in Corollary 3.2.

Gradient-SVD. Suppose the input second moment is isotropic in the full input space:

$$X^\top X = cI_d, \qquad c > 0. \tag{39}$$

Let $V_{G,r}$ contain the leading right singular vectors of G, with singular values $\Sigma_{G,r}$. The columns of $XV_{G,r}/\sqrt{c}$ are orthonormal eigenvectors of $R^\top R$, with eigenvalues $\eta_{\mathrm{loc}}^2 c \Sigma_{G,r}^2$. They span its leading rank-r eigenspace. The Gradient-SVD reader $A_{\mathrm{grad}} = V_{G,r}^\top$ exposes this same response subspace through $A_{\mathrm{grad}} X^\top = (XV_{G,r})^\top$. For rank-deficient X, the weaker condition $X^\top X = cP_X$ gives the corresponding result for the projected gradient GP_X. It applies directly to the raw-gradient construction when $G = GP_X$.

These are response-subspace identities, with coordinate scale left free. For the aligned orthonormal response frames above, the minimum-norm reader is $\Sigma_{X,r}^{-1} Q_r^\top$ in the activation case and $c^{-1/2} V_{G,r}^\top$ in the gradient case. Thus the unweighted singular-vector readers and the fixed-frame minimum-norm realizations can differ as matrices while exposing the same response subspace. Nonzero scalar scaling preserves that subspace. The ridge pullback in § 4.1 rescales the retained axes in these isotropic cases; on general probes it can also change the response subspace. Historical filtering acts afterward and can change which response directions remain accessible. § 5.2 evaluates the installed constructions on measured probes under common scaling and filtering rules.

Response on a second probe. For a fixed-frame reader $A = A^{\mathrm{me}} + N(I_d - P_X)$, the extension is silent on the matched probe because $N(I_d - P_X)X^\top = 0$. On another probe X_j, evaluated at the same fixed model state, its writer-mediated contribution is

$$B(A - A^{\mathrm{me}})X_j^\top = BN(I_d - P_X)X_j^\top, \tag{40}$$

which can be nonzero when $B \neq 0$ and $(I_d - P_X)X_j^\top \neq 0$. This extension vanishes when the matched activation matrix has full column rank, $\mathrm{rank}(X) = d$.

A.5 Proof of Unprotected First-Order Task Sensitivity

Proof of Corollary 3.3. Set $P = P_A = A^\dagger A$ and $U = G_t(I_d - P)$. Suppose first that $U = 0$. Then

$$G_t = G_t P = (G_t A^\dagger)A =: CA. \tag{41}$$

Taking $B = -\eta_{\mathrm{loc}} C$ gives $R = -\eta_{\mathrm{loc}} G_t X^\top = BAX^\top$. Hence $\mathcal{E}_R(A) = 0$; moreover, $\mathrm{rank}(R) \leq r$, so $\mathcal{E}_r^\star(R) = 0$ and $\mathcal{D}_{R,X}(A) = 0$, a contradiction. Therefore $U \neq 0$.

Let $\sigma_1 u_1 v_1^\top$ be a leading singular component of U. Because $UP = 0$, the nonzero right-singular vector satisfies $v_1^\top P = 0$. For any $\rho > 0$, set $\Delta W = \rho u_1 v_1^\top$. Then $\mathrm{rank}(\Delta W) = 1$, $\Delta W P = 0$, and $\|\Delta W\|_F = \rho$. Finally,

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\epsilon}\mathcal{L}_t(W_t + \epsilon \Delta W)\Big|_{\epsilon=0} &= \langle G_t, \Delta W \rangle_F \\ &= \langle U, \Delta W \rangle_F = \rho \sigma_1(U) > 0, \end{aligned} \tag{42}$$

which proves the claim. Q.E.D.

Positive reader deficit identifies unprotected task sensitivity. A nonzero $G_t(I_d - P_A)$ can also occur at zero deficit when it is invisible on the matched activation probe or coexists with reader coordinates that attain the optimal rank-r response fit on that probe.

Historical-union boundary. Corollary 3.3 uses $\mathrm{row}(A)$ as the protected boundary. For a later task $k > t$, let $P_{<k}^A := A_{<k}^\dagger A_{<k}$ project onto the exact row span of the stacked historical readers. If the loss gradient of old task t is re-evaluated at that later model state and denoted by $G_t^{(k)}$, its residual sensitivity is $G_t^{(k)}(I_d - P_{<k}^A)$. The union test uses the later-state gradient and all retained readers, updating both quantities in the original fixed-state pair (G_t, A_t). The implemented boundary in Eq. 12 uses $\widehat{P}_{<k}^{\mathrm{QR}}$, which may include QR completion directions when the historical stack is rank-deficient; its residual sensitivity is $G_t^{(k)}(I_d - \widehat{P}_{<k}^{\mathrm{QR}})$.

A.6 Proof of the Gram-Preconditioned Writer Update

Proof of Corollary 3.4. For fixed R, X, A, define $f_A(B) := \frac{1}{2}\|R - BZ_A\|_F^2$ and $K_Z := Z_A Z_A^\top$. Differentiation gives $\nabla_B f_A(B) = (BZ_A - R)Z_A^\top$. The SVD identity $Z_A^\top K_Z^\dagger = Z_A^\dagger$ holds at any rank, so the update direction can be written as the negative Gram-preconditioned gradient:

$$\Delta B_A := -\nabla_B f_A(B) K_Z^\dagger = (R - BZ_A)Z_A^\dagger. \tag{43}$$

Let $P := Z_A^\dagger Z_A$ be the orthogonal projector onto $\mathrm{row}(Z_A)$. Since $Z_A P = Z_A$ and $B^\star(A)Z_A = RP$, the direction satisfies $\Delta B_A Z_A = RP - BZ_A$: it corrects the current response toward the optimal response attainable with the reader. Substituting into Equation 10 gives

$$B^+ Z_A = BZ_A + (R - BZ_A)P = RP. \tag{44}$$

Theorem 3.1 identifies the writer gap as $f_A(B) - \min_{\bar{B}} f_A(\bar{B}) = \frac{1}{2}\|BZ_A - RP\|_F^2$. Using Equation 44 therefore yields

$$f_A(B^+) - \min_{\bar{B}} f_A(\bar{B}) = \frac{1}{2}\|B^+ Z_A - RP\|_F^2 = 0. \tag{45}$$

Thus the unit-step update achieves the optimal fitted response RP. These relations hold for rank-deficient Z_A, including $Z_A = 0$. The writer itself is $B^+ = B^\star(A) + B(I - Z_A Z_A^\dagger)$, which retains the component invisible to the probe response. If Z_A has full row rank, this component vanishes and $B^+ = B^\star(A)$. Q.E.D.

A.7 Coordinate Equivariance of Ideal Writer Conditioning

For an invertible $Q \in \mathbb{R}^{r\times r}$, let $Z_Q = QZ$, $B_Q = BQ^{-1}$, and consider a differentiable local loss $\mathcal{L}(B, Z) = \ell(BZ)$, with $G_B = \nabla_B \mathcal{L}(B, Z)$. For a common writer step size $\alpha > 0$, plain Euclidean descent is generally coordinate-sensitive: one step in the two equivalent realizations changes their realized responses by $-\alpha G_B(Q^\top Q - I)Z$.

Proposition A.2 (Ideal coordinate-equivariant writer preconditioning)**.** *Assume $K_Z \succ 0$ and define*

$$\mathcal{U}_Z(B) := B - \alpha G_B K_Z^{-1}. \tag{46}$$

For every invertible Q, let $K_{Z_Q} = QK_ZQ^\top$. Then

$$\mathcal{U}_{Z_Q}(B_Q) = \mathcal{U}_Z(B)Q^{-1}. \tag{47}$$

Hence the two updated coordinate realizations represent the same next low-rank response.

Proof of Proposition A.2. At equivalent coordinates, $G_{B_Q} = G_B Q^\top$ and

$$K_{Z_Q}^{-1} = (QK_ZQ^\top)^{-1} = Q^{-\top}K_Z^{-1}Q^{-1}. \tag{48}$$

Substitution gives

$$\begin{aligned}\mathcal{U}_{Z_Q}(B_Q) &= BQ^{-1} - \alpha G_B Q^\top Q^{-\top}K_Z^{-1}Q^{-1} \\ &= (B - \alpha G_B K_Z^{-1})Q^{-1} = \mathcal{U}_Z(B)Q^{-1}.\end{aligned} \tag{49}$$

Q.E.D.

This exact equivariance holds for the untruncated Euclidean update $B \leftarrow B - \alpha G_B K_Z^{-1}$. § 4.2 uses its geometry to motivate gradient preconditioning, implemented with a static activation moment, spectral truncation, accumulation, clipping, and AdamW. § 5.3 evaluates the contribution of AdamW-state reset under this training rule.

B InheritLoRA Implementation Details

Algorithm 1 gives the GradResp implementation of InheritLoRA used in the main experiments. Reader construction and Gram geometry instantiate the two principles; optimizer-state reset is specified separately in Appendix B.3.

Algorithm 1 InheritLoRA with GradResp for task t: fixed reader and writer optimizer-state reset.

Input: current-task data D_t, frozen base model, and frozen historical blocks.
Output: retained task block (A_t, B_t).

1: **Initialize:** obtain $\widetilde{A}_t^{(0)}$ from a base-model probe using Equation 11.
2: $A_t \leftarrow \mathrm{Hist}_{<t}\big(\mathrm{Scale}(\widetilde{A}_t^{(0)})\big)$, $B_t \leftarrow 0$; hold A_t fixed.
3: Collect C_t; set $H_t \leftarrow (A_tC_tA_t^\top)^\dagger_{;\tau}$.
4: **for each** writer optimizer update **do**
5: Accumulate $\widetilde{g}_{B_t}^{\mathrm{CE}} = g_{B_t}^{\mathrm{CE}}H_t$ over the prescribed micro-batches, clip to obtain $u_{t,s}$, and update B_t by Equation 54.
6: Reset the current writer's AdamW step counter and moment estimates; retain B_t and continue the global step and learning-rate schedule.
7: **end for**
8: Freeze (A_t, B_t), append it to the historical blocks, and retain A_t for subsequent historical filtering.

B.1 Response-Supported Reader Construction

For $\lambda > 0$, the solution to Equation 11 at task initialization has the closed form

$$\widetilde{A}_t^{(0)} = V_r^\top X\left(X^\top X + \lambda I\right)^{-1}. \tag{50}$$

The inverse is implemented by a linear solve. For fixed probe quantities X, V_r, $\widetilde{A}_t^{(0)} \to V_r^\top(X^\top)^\dagger = A^{\mathrm{me}}$ as $\lambda \downarrow 0$. The installed reader incorporates finite ridge, scaling, and historical filtering. The model retains fixed rank r; for response rank $q < r$, the nonzero modes provide

the active response coordinates. Both backbones extract the response coordinates by a direct SVD. T5 uses a primal $d \times d$ ridge system; LLaMA uses the equivalent sample-side ridge system formed from the n activation rows retained for the current-task probe.

Before installation, the raw proposal is rescaled by

$$\mathrm{Scale}(A) := \frac{1/\sqrt{3d}}{\mathrm{sd}(A) + \epsilon_{\mathrm{num}}} A, \qquad \epsilon_{\mathrm{num}} = 10^{-12}, \tag{51}$$

where $\mathrm{sd}(A)$ is the sample standard deviation of all rd entries. This multiplicative scaling preserves the relative values of the entries. Initialization installs $A_t^{(0)} = \mathrm{Hist}_{<t}(\mathrm{Scale}(\widetilde{A}_t^{(0)}))$, which is then held fixed during all writer updates for the current task.

B.2 Reader-Conditioned Writer Geometry

In Equation 14 and Equation 15, $K_t, H_t \in \mathbb{R}^{r\times r}$ and $g_{B_t}^{\mathrm{CE}} \in \mathbb{R}^{m\times r}$. On both backbones, the static moment C_t is an unnormalized sum of outer products over the retained activation rows. T5 adds the outer-product sums of its collected micro-batches; LLaMA stores the selected rows and forms K_t from their reader-projected coordinates. The moment is collected after installing the task-start reader and remains fixed throughout the task. With a fixed reader, H_t is computed once and cached. Upstream adaptation can change subsequent activations, while the cached geometry remains tied to the task-start sample. The backbone-specific learning rates and clipping thresholds are independent of the collected token count.

For $K_t = V\,\mathrm{diag}(\mu_i)V^\top$, negative numerical eigenvalues are clamped to zero. For $\tau > 0$, define

$$\phi_\tau(\mu_i) = \begin{cases} \mu_i^{-1}, & \mu_{\max} > 0 \text{ and } \mu_i \geq \tau\mu_{\max}, \\ 0, & \text{otherwise}, \end{cases} \qquad K_{t;\tau}^\dagger = V\,\mathrm{diag}(\phi_\tau(\mu_i))\,V^\top. \tag{52}$$

The reconstruction is symmetrized, with zero inverse weight assigned to unsupported directions and weak modes removed by the spectral cutoff.

B.3 Per-Update Optimizer-State Reset

Reset makes every update the first AdamW step for its moment estimates, while the writer weights and global training step continue across updates. For one writer parameter matrix, let m, v denote the first- and second-moment buffers and k the per-parameter step counter. Initially, and after each reset, $(m, v, k) = (0, 0, 0)$. Let $u_{t,s} \in \mathbb{R}^{m\times r}$ denote the optimizer-input gradient at update s, after Gram preconditioning, micro-batch accumulation, and gradient clipping. The next bias-corrected AdamW update therefore uses

$$\begin{aligned} &k = 1, && m = (1-\beta_1)u_{t,s}, && v = (1-\beta_2)u_{t,s}^{\odot 2}, \\ &\widehat{m} = \frac{m}{1-\beta_1} = u_{t,s}, && \widehat{v} && = \frac{v}{1-\beta_2} = u_{t,s}^{\odot 2}. \end{aligned} \tag{53}$$

Here $\beta_1, \beta_2 \in [0, 1)$ are the moment-decay coefficients and $\odot 2$ denotes elementwise squaring. With AMSGrad disabled and zero weight decay, substituting these moments into AdamW yields

$$B_t^{(s+1)} = B_t^{(s)} - \eta_s \frac{u_{t,s}}{|u_{t,s}| + \epsilon_{\mathrm{Adam}}}. \tag{54}$$

Here $\sqrt{\widehat{v}} = |u_{t,s}|$, and absolute value and division act elementwise; η_s is the scheduled learning rate and $\epsilon_{\mathrm{Adam}} > 0$ is AdamW's denominator stabilizer. After each update, reset clears the per-parameter optimizer state and preserves B_t and the global learning-rate schedule.

Implementation equivalence check. We compare the archived T5 reset callback with the explicit update in Equation 54 on identical prescribed gradient sequences. CPU tests cover 54 configurations and 594 optimizer updates, including single- and multi-step sequences, zero and small gradients, different moment coefficients, and a varying learning-rate schedule. The maximum absolute parameter discrepancy is 7.45×10^{-9} in FP32 and 1.39×10^{-17} in FP64. Of these configurations, six use FP32 writer parameters from three saved T5 layers and exercise the original preconditioner, four-way gradient accumulation, and clipping; their maximum discrepancy is 2.79×10^{-9}. At every reset, checks verify that the writer optimizer state is cleared while parameter values, the global step, and the scheduler state are unchanged.

B.4 Historical Reader Boundary

Equation 12 defines the historical filter used in Equation 13. Historical readers are stacked by rows, and reduced QR of $A_{<t}^{\top}$ produces a column basis $Q_{<t}$ with $\widehat{P}_{<t}^{\mathrm{QR}} = Q_{<t}Q_{<t}^{\top}$. Empty histories or stacks with Frobenius norm below ϵ_{num} use the zero projector. Projection is performed without explicitly forming the full projector. For a reference signal S, row restoration uses

$$[\mathcal{R}_S(Y)]_{i,:} = \frac{\max(\|S_{i,:}\|_2, \epsilon_{\mathrm{num}})}{\max(\|Y_{i,:}\|_2, \epsilon_{\mathrm{num}})} Y_{i,:}. \tag{55}$$

We use reduced QR without numerical rank truncation, so an exactly rank-deficient historical stack can introduce completion directions beyond its strict row span.

B.5 Current-Task Data Interfaces

Reader construction uses a matched probe on the frozen base model without historical adapters; writer geometry uses a separate activation collection on the continual model after installing the current reader. Both use current-task training records. Tab. 6 summarizes their sampling rules; an activation row is one token position at an adapted layer.

Table 6: Current-task data interfaces.

Interface	T5-large	LLaMA 3.1–8B
Reader construction: matched gradient and activation probe		
Training records	Up to 512	Up to 512
Record selection	Seeded permutation prefix	Seeded permutation prefix
Physical micro-batch	4	1
Activation rows	All valid token rows	At most 256 valid token rows
Probe model	Frozen base model	Frozen base model
Gradient targets	Labeled target sequences	Labeled target sequences
Writer geometry: task-start activation collection		
Training records	First up to 256	First up to 64
Physical micro-batch	8	1
Activation rows	Stop after the micro-batch reaching 8192 rows	At most 256 valid token rows
Collection model	Continual model after reader installation	Continual model after reader installation

Supervised probe and row selection. Teacher forcing conditions each target-token prediction on the labeled ground-truth prefix. We collect X and backpropagate the same supervised forwards to obtain G, averaging per-example target-token mean losses over the selected records while holding model parameters fixed. The virtual dense step in Equation 3 defines the response reference. LLaMA selects the integer-rounded-down midpoint of each equal-width interval in the complete valid-token stream, for both interfaces; its 256-row budget is distinct from sequence-length limits. T5 accumulates complete micro-batch outer-product sums, so the final micro-batch can take the accumulated row count above the nominal 8192-row threshold.

Implementation note. The archived T5 runs reported in this paper use encoder or decoder masks selected by module location during static geometry collection. Masks are applied when their dimensions match the layer input; otherwise input positions are flattened and can include padding. This includes decoder cross-attention value inputs when their source length differs from the decoder-mask length. Reader initialization instead assigns the encoder mask explicitly to those inputs. Optimization settings are specified in Appendices C.1 and C.2.

B.6 Implementation-Level Computational Complexity

We count dense linear-algebra operations per adapted $m \times d$ layer, with rank r, n_C static rows, and $h = (t-1)r \leq d$ historical reader rows, as in the reported streams. The counts below cover historical filtering, writer-geometry setup, gradient preconditioning, and retained state. Model forward/backward passes and task-start response construction incur additional costs. The response SVD and ridge solve occur once per task with the sample budgets in Appendix B.5.

Installation and writer geometry. Constructing the historical QR basis costs $O(dh^2)$ time and $O(dh)$ storage; the task-start projection costs $O(rdh)$. Computing H_t once costs $O(rd^2+r^2d+r^3)$ for T5's stored $d \times d$ moment, versus $O(n_C dr + n_C r^2 + r^3)$ for LLaMA's stored rows; static storage is $O(d^2)$ and $O(n_C d)$, respectively. Applying the cached preconditioner requires $O(mr^2)$ operations for each $m \times r$ writer-gradient matrix.

Persistent reader–writer factors require $O(tr(m+d))$ coefficients per layer; current-writer optimizer buffers require $O(mr)$ storage. With unmerged historical factors, the adapter contribution to a forward pass is $O(tr(m+d))$ per token.

C Experimental Protocol and Benchmark Provenance

This appendix specifies the local protocols, task orders, evaluation metrics, and sources of the published benchmark values. Appendix D collects the supplementary measurements and diagnostics that support the main experimental results.

C.1 T5-Large Protocol

Model and optimization. All local T5 runs start from the same frozen T5-large checkpoint. LoRA is applied to the query and value projections of encoder self-attention, decoder self-attention, and decoder cross-attention, giving 144 adapted matrices. We use rank 8, scale 32, dropout 0.1, maximum source and target lengths of 512 and 50, physical batch size 16, gradient accumulation 4, and AdamW with learning rate 10^{-3}, zero weight decay, a constant schedule, and global gradient clipping at 1. Inputs retain the textual task and dataset prefixes used by the protocol. A single cumulative adapter serves all tasks through a shared inference path. Evaluation uses greedy generation of at most 50 new tokens. Accuracy is exact match after lowercasing, deleting punctuation, and collapsing whitespace.

InheritLoRA uses at most 512 current-task examples to construct the task-start reader and 256 examples to estimate the writer moment. The reader remains fixed within each task. The 15-task reader intervention and fixed-reader controls use O4, one epoch per task, and seeds 42, 43, and 44. The reader ridge coefficient is $\lambda = 10^{-4}$, and the spectral cutoff is $\tau = 10^{-2}$. Appendices D.3 and D.4 specify the reader constructions and writer controls, respectively.

Historical access. Training and auxiliary probes use current-task examples. Persistent historical information consists of frozen low-rank model states, which define the historical reader boundary.

C.2 LLaMA 3.1–8B Protocol

The LLaMA main experiments use rank 8, scale 32, dropout 0.1, and one epoch per task, with a physical batch size of 1 and gradient accumulation of 32. AdamW uses learning rate 10^{-4} and zero weight decay. Maximum source and target lengths are 512 and 50; evaluation batch size is 8. Task-start reader construction uses up to 512 current-task training examples; writer geometry uses up to 64 examples, with a token-row budget of 256. The reader ridge coefficient is 10^{-4} and the spectral cutoff is 10^{-2}. As in the T5 main runs, task-start reader probes use the base model, the reader is fixed within each task, and only the current writer is trained using Gram preconditioning and per-update AdamW-state reset. Historical adapters remain frozen.

C.3 Task Orders, Metrics, and Aggregation

Packaged data and length limits. The 15-task T5 experiments train on all records in each packaged benchmark training split, whose sizes are listed in Tab. 7. These splits are preselected subsets of the upstream datasets. Probe budgets apply to auxiliary reader and geometry construction; writer training uses each packaged training split in full. T5 retains a prefix of at most 512 source tokens and 50 target tokens. Probe records are drawn from the same current-task training split and may overlap between reader initialization and writer-geometry construction.

Task-order matching. We match published results by the complete task sequence in Tab. 8, accounting for differences in order labels across source papers.

Table 7: Record counts in the packaged training and test splits of the benchmark.

Task	Train	Test	Task	Train	Test
BoolQA	2,000	3,270	SST-2	2,000	872
COPA	400	100	Amazon	5,000	7,600
MultiRC	2,000	4,848	Yelp	5,000	7,600
CB	250	56	AG News	4,000	7,600
MNLI	3,000	7,600	DBpedia	14,000	7,600
RTE	2,000	277	Yahoo	10,000	7,600
QQP	2,000	7,600	WiC	2,000	638
IMDB	2,000	7,600			

For every task, formal runs train on all records included in that task's packaged training split.

Table 8: Standard (O1–O3) and Long-Sequence (O4–O6) task orders.

Order	Task sequence
O1	DBpedia → Amazon → Yahoo → AG News
O2	DBpedia → Amazon → AG News → Yahoo
O3	Yahoo → Amazon → AG News → DBpedia
O4	MNLI → CB → WiC → COPA → QQP → BoolQA → RTE → IMDB → Yelp → Amazon → SST-2 → DBpedia → AG News → MultiRC → Yahoo
O5	MultiRC → BoolQA → WiC → MNLI → CB → COPA → QQP → RTE → IMDB → SST-2 → DBpedia → AG News → Yelp → Amazon → Yahoo
O6	Yelp → Amazon → MNLI → CB → COPA → QQP → RTE → IMDB → SST-2 → DBpedia → AG News → Yahoo → MultiRC → BoolQA → WiC

Published order labels are mapped to the complete sequences listed here.

C.3.1 Acquisition, Final Accuracy, and Backward Transfer

Let $a_{s,t}$ be held-out accuracy on task t after learning tasks $1, \ldots, s$, with the evaluation interface fixed across checkpoints. For a stream of $T > 1$ tasks, the four primary metrics are

$$\mathrm{LC} = \frac{1}{T}\sum_{t=1}^{T} a_{t,t}, \qquad \mathrm{FinalAcc} = \frac{1}{T}\sum_{t=1}^{T} a_{T,t}, \tag{56}$$

$$\mathrm{AvgForget} = \frac{1}{T-1}\sum_{t=1}^{T-1} \max\{a_{t,t} - a_{T,t}, 0\}, \qquad \mathrm{BWT} = \frac{1}{T-1}\sum_{t=1}^{T-1} (a_{T,t} - a_{t,t}). \tag{57}$$

Average learning accuracy, denoted LC throughout this paper, measures task acquisition immediately after learning. The same quantity is called AvgLAcc by Chen et al. (2023) and LA by Graldi et al. (2025). FinalAcc measures the accuracy retained at the end of the stream; BWT follows the signed backward-transfer definition of Lopez-Paz & Ranzato (2017). AvgForget counts only decreases from each task's acquisition score, while BWT also counts improvements on earlier tasks. Table headers use Final ACC and Avg. forget. for FinalAcc and AvgForget, respectively.

Cancelling the final task's $a_{T,T}$ term gives

$$\mathrm{BWT} = \frac{T}{T-1}\big(\mathrm{FinalAcc} - \mathrm{LC}\big). \tag{58}$$

Thus BWT expresses the signed acquisition–retention gap determined by LC and FinalAcc. AvgForget averages positive task-level declines. Reporting LC alongside forgetting identifies the acquisition level from which retention is measured.

C.3.2 Peak-Normalized Forgetting

Our AvgForget uses the score immediately after a task is learned as its reference. For the reader-only intervention, we also report peak-normalized forgetting (CFr) in Table 17. Following the historical-best reference of Chaudhry et al. (2018), let $h_t = \max_{t \le s < T} a_{s,t}$ denote a task's highest accuracy

from acquisition to the stage before the final evaluation. CFr normalizes the loss by this peak (Graldi et al., 2025):

$$\mathrm{CFr} = \frac{1}{T-1}\sum_{t=1}^{T-1}\frac{h_t - a_{T,t}}{h_t}, \qquad h_t > 0 \text{ for all } t < T. \tag{59}$$

CFr measures the fraction of peak accuracy lost. Together with LC and final per-task scores, it helps distinguish small absolute losses after weak acquisition from strong retention after successful acquisition. A negative term indicates that final accuracy exceeds the earlier peak. All historical peaks in the reported intervention are positive. We report CFr alongside raw accuracy to show both the normalized loss and its reference scale.

C.3.3 Aggregation and Reporting

Each metric, including the supplementary CFr, is computed within a run before aggregation. The 15-task reader intervention and fixed-reader controls use equal-weighted means over O4 seeds 42–44. Other diagnostics retain the aggregation stated at the start of their subsections.

Comparisons use matched task orders, evaluation checkpoints, task weights, reference scores, and sign/clipping conventions. Peak-referenced metrics use the recorded task-boundary evaluations. Accuracies are reported as percentages and their absolute differences in percentage points; CFr is dimensionless and may be displayed as a percentage. Held-out evaluation records are kept separate from the data used for training and auxiliary probes. Main benchmark tables retain their one-decimal display. Detailed local accuracy, forgetting, backward-transfer, and percentage-point effects use two decimals; correlations use three, and NLL uses up to four significant digits. Small numerical residuals use scientific notation. Source-reported results and baseline summaries available only as rounded values retain their supplied precision. Means and contrasts are computed from unrounded values before display rounding.

C.4 Source-Reported Benchmark Provenance

Tab. 9 maps the external rows in Tables 2 and 3 to their reporting papers and source tables. Primary-source values take precedence; starred rows are third-party reimplementations. The InheritLoRA rows use our fixed-reader runs in Appendix D.1; Appendix C.5 separately reports the results from our local baseline implementations.

Source-specific conventions. For LLaMA SPARTA+Replay, Table 10 reports O1–O3 as $(80.6, 81.0, 80.7)$ and O4–O6 as $(83.1, 81.7, 81.8)$. Their means, 80.8% and 82.2%, agree with the final-performance (FP) values in Table 1. SPARTA's formal table contains the order-wise values previously reported as DATA; we list them once under SPARTA. Recurrent-KIF reports order-averaged OP with the same endpoint definition as FinalAcc; its unreported order-wise cells remain blank.

Capacity and comparison scope. Fixed-capacity methods keep one shared adapter or aggregate state; growing methods retain task-indexed adapter or state blocks. These descriptions exclude the frozen backbone and optimizer buffers. Recurrent-KIF's LLaMA2 results use a different backbone from LLaMA 3.1–8B. Prompt-only comparison grids use oracle task identity and larger training budgets. Extended comparisons and their protocol differences appear in Appendix C.6.

C.5 Local Baseline Runs

Local implementations of SeqLoRA, IncLoRA, and O-LoRA. In Tab. 10, our implementations of SeqLoRA, IncLoRA, and O-LoRA (superscript L) use seeds 42–44, the packaged training and test splits, and the task orders specified above. All three cover LLaMA 3.1–8B; SeqLoRA and IncLoRA additionally cover T5-large. For all three baselines, the backbone is frozen, the LoRA rank is 8, $\alpha = 32$ (multiplicative scale $\alpha/r = 4$), and adapter dropout is 0.1. T5 adapts 144 query/value matrices across encoder self-attention, decoder self-attention, and cross-attention; LLaMA adapts the 64 query/value matrices in its 32 self-attention layers.

T5 uses AdamW with learning rate 10^{-3}, physical batch size 16, and 4 gradient-accumulation steps (effective batch size 64). LLaMA uses AdamW with learning rate 10^{-4}, physical batch size 1,

Table 9: Provenance and protocol of the main benchmark baselines.

Method	Backbone	Reporting source	Replay	Epochs	Update / retained state
SeqLoRA*, IncLoRA*	T5	Wang et al. (2023), Table 2	No	–	Shared / task-indexed LoRA
O-LoRA	T5	Wang et al. (2023), Table 2	No	1	Growing task LoRAs
SeqLoRA*, O-LoRA*	LLaMA	Liao et al. (2026), Table 10	No	–	Shared / task-indexed LoRA
CLoRA, CLoRA + O-LoRA	T5	Lu et al. (2025), Table 5	No	1	Constrained LoRA / task LoRAs
ASO-LoRA	T5	Wang et al. (2026), Table 1	No	1	Soft-orthogonal task LoRAs
GORP	T5	Wang et al. (2025a), Table 2	No	1†	Low-rank and projected full-rank updates
ELLA	Both	Das Biswas et al. (2026), Table 1	No	1	Accumulated adapter energy
SPARTA	Both	Liao et al. (2026), Table 10	No	1	Branched adapter state
SPARTA + Replay	Both	Liao et al. (2026), Table 10	Yes	–	Branches and old examples
Recurrent-KIF	T5	Feng et al. (2025), Table 1	Yes	10	Replay-assisted; aggregate OP only

T5: T5-large; LLaMA: LLaMA 3.1–8B. All rows retain the source protocols. A dash marks an epoch count not specified in this provenance summary. *Reported by another method's paper. †GORP describes one epoch; its released O5/O6 scripts assign two to MNLI.

and 32 accumulation steps (effective batch size 32), with bfloat16 model computation. Both use zero weight decay, a constant learning rate, no warmup, and global gradient clipping at 1. The configured training budget is one epoch per task. In the recorded LLaMA runs, incomplete terminal accumulation windows are omitted by the training loop; for example, CB processes 224 of 250 examples and COPA processes 384 of 400 examples. Maximum source length is 512 tokens and maximum target length is 50 tokens; evaluation batch sizes are 32 for T5 and 8 for LLaMA. LLaMA retains its native BOS/EOS tokens and uses EOS for padding; prompt and padding positions are excluded from the language-model loss. T5 accumulates microbatch token-mean losses, whereas LLaMA normalizes its task loss over the valid target tokens in each accumulation window.

SeqLoRA carries one rank-8 adapter across tasks and updates both factors. IncLoRA initializes a new rank-8 block at each task, freezes all historical blocks, and sums their contributions during inference. O-LoRA uses the same expanding architecture and adds the entrywise absolute overlap penalty $P = \sum_\ell \|A_{\text{old},\ell} A_{\text{new},\ell}^\top\|_1$, with configured coefficient 0.5 and no additional factor-norm penalty. In the recorded local implementation, $0.5P$ is added at every microbatch; with 32 accumulation steps, its accumulated pre-clipping gradient has coefficient 16 relative to the window-normalized task loss. For all three baselines, AdamW is recreated at task boundaries and its state is retained within each task. Both current factors are trained with AdamW on current-task data. Each task is evaluated after acquisition, and the complete test stream is evaluated after the final task using greedy generation and the same normalized exact-match metric as the main experiments. The local comparison table uses these final-stream accuracies.

Author-code O-LoRA reproduction. We additionally run the authors' released O-LoRA code (commit `07117e1`)[1] adapted to LLaMA 3.1–8B, marked with superscript A in Tab. 10. These runs cover O1–O6 and seeds 42–44 with the same data splits, rank, scale, dropout, learning rate, batch size, and one-epoch budget specified above. The adaptation uses LLaMA 3.1's native special tokens while retaining the original masked microbatch loss, joint prompt–answer truncation, and decoded-answer extraction. Training uses DeepSpeed ZeRO-2 with CPUAdam offload and the original WarmupLR scheduler, including its two initial zero-learning-rate steps. Joint scaling of the task loss and overlap penalty during accumulation gives an effective overlap coefficient of 0.5, compared with 16 in implementation L. These rows therefore compare complete training and evaluation

[1] https://github.com/cmnfriend/O-LoRA

Table 10: Local baseline runs with two O-LoRA implementations.

Method	Seed	O1	O2	O3	Avg	O4	O5	O6	Avg
T5 / SeqLoRA	42	64.8	61.5	73.3	66.5	54.5	43.1	22.4	40.0
T5 / SeqLoRA	43	53.0	57.8	69.7	60.2	40.6	40.2	34.1	38.3
T5 / SeqLoRA	44	54.4	64.5	66.9	61.9	40.1	44.4	33.5	39.3
T5 / SeqLoRA	42–44	57.4	61.3	70.0	62.9	45.0	42.5	30.0	39.2
T5 / IncLoRA	42	72.8	67.2	75.2	71.7	62.6	54.2	66.7	61.1
T5 / IncLoRA	43	77.1	76.9	72.6	75.6	62.7	56.0	63.5	60.7
T5 / IncLoRA	44	72.2	66.4	74.0	70.9	62.2	53.4	63.4	59.7
T5 / IncLoRA	42–44	74.1	70.2	73.9	72.7	62.5	54.5	64.5	60.5
T5 / GORP	42	75.9	73.5	76.8	75.4	66.4	63.1	71.2	66.9
T5 / GORP	43	76.3	75.0	73.8	75.1	68.9	65.5	72.1	68.8
T5 / GORP	44	77.1	69.8	74.0	73.7	67.3	64.5	72.9	68.2
T5 / GORP	42–44	76.5	72.8	74.9	74.7	67.5	64.4	72.1	68.0
T5 / ASO-LoRA	42	78.1	77.8	77.9	78.0	69.8	65.2	73.0	69.3
T5 / ASO-LoRA	43	78.6	78.7	76.7	78.0	67.2	69.6	68.3	68.4
T5 / ASO-LoRA	44	78.1	77.5	74.5	76.7	66.8	64.5	67.0	66.1
T5 / ASO-LoRA	42–44	78.3	78.0	76.4	77.6	67.9	66.4	69.4	67.9
LLaMA / SeqLoRA	42	80.1	81.0	79.2	80.1	72.5	69.4	80.6	74.2
LLaMA / SeqLoRA	43	80.9	79.6	79.7	80.1	74.2	70.6	81.0	75.3
LLaMA / SeqLoRA	44	79.7	80.9	80.1	80.2	74.5	71.9	82.0	76.1
LLaMA / SeqLoRA	42–44	80.2	80.5	79.7	80.1	73.7	70.6	81.2	75.2
LLaMA / IncLoRA	42	80.3	80.8	81.8	81.0	78.7	75.3	82.3	78.8
LLaMA / IncLoRA	43	81.5	81.2	81.6	81.4	79.4	78.6	83.2	80.4
LLaMA / IncLoRA	44	81.0	81.2	81.1	81.1	79.8	75.8	82.7	79.4
LLaMA / IncLoRA	42–44	81.0	81.1	81.5	81.2	79.3	76.6	82.7	79.5
LLaMA / O-LoRA^{L}	42	80.7	81.2	81.2	81.0	80.8	81.5	80.6	81.0
LLaMA / O-LoRA^{L}	43	80.8	81.3	81.5	81.2	81.5	80.7	81.8	81.3
LLaMA / O-LoRA^{L}	44	81.1	80.6	81.3	81.0	80.3	79.1	80.5	80.0
LLaMA / O-LoRA^{L}	42–44	80.8	81.1	81.3	81.1	80.9	80.4	81.0	80.8
LLaMA / O-LoRA^{A}	42	72.6	72.3	72.4	72.4	67.8	66.9	67.0	67.2
LLaMA / O-LoRA^{A}	43	71.9	71.1	72.4	71.8	68.4	63.5	65.7	65.9
LLaMA / O-LoRA^{A}	44	70.6	71.8	70.5	71.0	69.4	68.0	68.0	68.5
LLaMA / O-LoRA^{A}	42–44	71.7	71.7	71.8	71.7	68.5	66.1	66.9	67.2
LLaMA / SPARTA	42	80.9	73.1	79.1	77.7	77.6	72.9	80.0	76.8
LLaMA / SPARTA	43	76.2	75.9	78.8	76.9	75.7	71.6	76.8	74.7
LLaMA / SPARTA	44	79.6	81.1	80.9	80.6	80.3	75.2	76.9	77.5
LLaMA / SPARTA	42–44	78.9	76.7	79.6	78.4	77.9	73.2	77.9	76.3

T5 denotes T5-large; LLaMA denotes LLaMA 3.1–8B. All entries are our own runs. O-LoRA^{L}: our local implementation; O-LoRA^{A}: authors' code adapted to LLaMA 3.1–8B. Values are final accuracies (%). Rows report individual seeds or three-seed means (42–44).

pipelines. Seventeen streams ran on A800 GPUs; O4/seed 44 ran on an RTX 5090 with the same adapted source and verified data and model files. All tasks are evaluated only after the final task, using greedy generation with evaluation batch size 4 and a 50-token generation limit.

Local baseline results. Tab. 10 gives seed-level results for our SeqLoRA, IncLoRA, O-LoRA, GORP, ASO-LoRA, and SPARTA runs. Each available order cell is the mean test accuracy over tasks after the complete stream. Each Avg equally weights all three corresponding orders for the same seed and is reported for complete three-order groups. Rows labeled 42–44 report equally weighted means over the results from the three training seeds.

Local GORP protocol. Our T5-large GORP implementation uses seeds 42–44, rank 8, $\alpha = 32$, adapter dropout 0.1, one epoch per task, and effective batch size 64. The LoRA learning rate is 10^{-3}; the full-rank learning rate is 10^{-5} for O1–O3 and 10^{-4} for O4–O6, with projection scales 1 and 0.25,

respectively. The projection rank is 8 and the update gap is 10. The checkpoint chain retains all 48 trainable full-rank matrices and loads historical gradient subspaces from the preceding checkpoint. The loss uses the released parameter-name selection, under which the configured $\lambda_3 = 0.05$ term contributes zero. Final accuracies use every task's complete test split after training the full stream.

Local ASO-LoRA protocol. Our T5-large ASO-LoRA runs use seeds 42–44, rank 8, $\alpha = 32$, adapter dropout 0.1, and one epoch per task. Training uses AdamW with learning rate 10^{-3}, zero weight decay, constant scheduling, bfloat16 computation, physical batch size 4, and 16 gradient-accumulation steps (effective batch size 64). The configured regularization coefficients are $\lambda_1 = 0.5$ and $\lambda_2 = 0$. Attribution comparison uses Spearman correlation, five interpolation points, and an interval of 50 training microbatches; comparison starts with the second task. Source and target limits are 512 and 50 tokens. Final accuracies average the complete test sets of all tasks in each stream, with greedy generation.

Local SPARTA protocol. Our LLaMA SPARTA reimplementation uses the common packaged splits and native-tokenizer evaluation interface, with seeds 42–44. It retains the released attention and MLP adaptation scope (224 projections), ranks 2/8, orthogonality coefficient 10, eight weight components, and layer gap 4. Training uses AdamW, learning rate 10^{-4}, constant scheduling, zero weight decay, gradient clipping at 1, bfloat16 computation, physical batch size 1, and 32 accumulation steps, for one configured epoch per task. Source and target limits are 512 and 50 tokens. Stochastic restoration is applied every 200 optimizer updates with probability 10^{-5}, using the task-start branch weights. Optimizer states are retained within tasks. Final evaluation uses task-agnostic routing over all learned components, greedy generation, and normalized exact match.

C.6 Extended Published Comparison

Tab. 11 extends the T5-large Long-Sequence comparison, reporting accuracy and the historical information retained by each method.

Table 11: T5-large comparison context and InheritLoRA results.

Method	Historical information	O4 ↑	O5 ↑	O6 ↑	Mean ↑
SeqFT (Wang et al., 2023)	none	7.4%	7.4%	7.5%	7.4%
SeqLoRA (Wang et al., 2023)	none	2.3%	0.6%	1.9%	1.6%
IncLoRA (Wang et al., 2023)	frozen task LoRAs	63.3%	58.5%	61.7%	61.2%
SeqSVD (Qiao & Mahdavi, 2024)	none	13.7%	13.8%	12.2%	13.2%
EWC (Wang et al., 2023)	parameter importance	45.3%	44.5%	45.6%	45.1%
LwF (Wang et al., 2023)	previous teacher outputs	50.1%	43.1%	47.4%	46.9%
L2P (Wang et al., 2023)	prompt pool	57.5%	53.8%	56.9%	56.1%
L-CL (Qiao & Mahdavi, 2024)	low-rank history	66.5%	64.0%	69.0%	66.5%
B-CL (Qiao & Mahdavi, 2024)	low-rank history	65.7%	66.4%	69.2%	67.1%
IncSVD (Qiao & Mahdavi, 2024)	frozen task factors	67.6%	65.3%	62.6%	65.2%
LB-CL (Qiao & Mahdavi, 2024)	low-rank history	68.4%	67.3%	71.8%	69.2%
O-LoRA (Wang et al., 2023)	frozen task LoRAs	72.3%	64.8%	71.6%	69.6%
O-LoRA + MIGU (Du et al., 2024)	LoRAs and update mask	67.3%	68.5%	74.2%	70.0%
SPARTA (Liao et al., 2026)	branched adapter state	71.5%	70.5%	68.0%	70.0%
SPARTA + Replay (Liao et al., 2026)	old examples and branches	75.6%	73.2%	74.1%	74.3%
Recurrent-KIF (Feng et al., 2025)	replay-assisted	–	–	–	77.8%
ELLA (Das Biswas et al., 2026)	accumulated adapter energy	73.4%	72.0%	75.4%	73.6%
CLoRA (Lu et al., 2025)	constrained LoRA	73.6%	66.4%	72.4%	70.8%
CLoRA + O-LoRA (Lu et al., 2025)	constrained task LoRAs	72.0%	67.3%	77.3%	72.2%
ASO-LoRA (Wang et al., 2026)	soft-orthogonal task LoRAs	73.7%	67.2%	77.4%	72.8%
N-LoRA (Yang et al., 2025)	sparse task LoRAs	73.6%	70.3%	73.2%	72.4%
GORP (Wang et al., 2025a)	low- and projected full-rank updates	76.1%	76.2%	75.6%	76.0%
OSFT (Muon-OGD rerun) (Nayak et al., 2026; Lu et al., 2026)	constrained high/full-rank update	71.6%	69.6%	72.7%	71.3%
InheritLoRA (ours)	frozen reader–writer blocks	76.4%	76.4%	74.9%	75.9%

Each external row cites its primary source table when available. InheritLoRA reports equal means over seeds 42–44 with one epoch per task; external rows retain their source protocols.

Recurrent-KIF Table 1 reports only order-averaged T5-large results, using OP with the same endpoint definition as FinalAcc: 78.4% OP and -2.8 pp BWT on the Standard benchmark, and 77.8% OP and -3.6 pp BWT on the Long-Sequence benchmark, each averaged across three orders. It uses replay and 10 epochs per task (source Appendix D); unreported order-wise cells remain blank.

The consolidated context includes different epoch budgets and update scopes. O-LoRA, CLoRA (with or without O-LoRA), ASO-LoRA, and the OSFT rerun use one epoch per task; N-LoRA uses ten. GORP's paper describes one epoch, while its released O5/O6 scripts assign two epochs to MNLI. GORP includes projected full-rank updates. The OSFT order-wise cells come from the Muon-OGD preprint's comparator rerun.

C.7 Single-Task Efficiency: Protocol and Implementations

Measurement scope. The efficiency columns in Tables 2 and 3 measure local implementations in a common training loop, separately from the source-reported continual accuracies. We use CB as a standalone first task: all 250 training examples, one epoch, seed 42, and effective batch size 8. T5-large uses physical batch 8; LLaMA 3.1–8B uses physical batch 1 with accumulation 8. Within each backbone, all methods use the same shuffled example order, microbatch contents, and inference prompts. There are 31 full optimizer updates and a final update on two examples, normalized by the actual accumulation-group size. No second task or intermediate evaluation is run. Each combination is measured once on one of four NVIDIA RTX 5090 GPUs, with exclusive use of that GPU during its run. The environment uses PyTorch 2.12.0+cu130 and Transformers 4.46.3, with two CPU threads and model compilation disabled for all timed runs.

Optimization and parameter counting. Ordinary LoRA branches use rank 8, scale parameter 32, dropout 0.1, and query/value targets: 144 matrices in T5-large and 64 in LLaMA 3.1–8B. Source and target length limits are 512 and 50. Both backbones use bf16 autocast; T5 base weights are float32 and LLaMA base weights are bf16, with float32 ordinary LoRA factors. The shared loop uses AdamW, constant learning rates of 10^{-3} (T5) and 10^{-4} (LLaMA), zero weight decay, and global gradient clipping at 1. GORP retains its projection optimizer, and SPARTA its native branched architecture. Trainable parameters, in millions (10^6), count the tensors updated by the task-loss optimizer, checked against its parameter groups. They exclude frozen backbone weights, readers, and regularization matrices. InheritLoRA also computes gradients of target backbone layers during reader construction; that work is timed in initialization. Appendix D.11 specifies the reader and writer state retained across the continual-learning sequence.

Timing boundaries. Training time is the arithmetic mean of the 26 full optimizer-step windows after excluding the first five updates and the final partial batch. CUDA synchronization brackets each window, which includes gradient clearing, input transfer to the GPU, forward/backward computation, clipping, the optimizer update, and method-specific update operations such as writer state reset. Training data are tokenized beforehand. Model loading, adapter installation, reader construction, and static Gram setup are outside the step windows. Using the mean includes periodic projection work such as GORP's, which a median can obscure.

Inference uses the same first eight CB test prompts within each backbone, without answers. After two warm-up repetitions, we time five greedy generation repetitions with batch 8, both minimum and maximum new-token counts set to 32, one beam, and the key/value cache enabled. Models are in evaluation mode and adapters remain unmerged. Throughput is the 1,280 generated tokens divided by total synchronized generation time across the five repetitions. Tokenization, input transfer, and output text decoding are excluded. Fixed output length controls for differences in answer length.

Implementation coverage. SeqLoRA and IncLoRA use the common LoRA layers without a historical block at the first task. O-LoRA adds the reader-overlap expression, whose historical factor has zero rows here. ASO-LoRA uses its first-task reduction without a previous model, so attribution comparisons are inactive. Recurrent-KIF uses its empty-memory first-task reduction: memory-dependent importance updates and outer fusion are inactive. These timings characterize first-task operations in the common training loop. End-to-end costs of the original training frameworks are outside this measurement.

CLoRA uses the released regularizer computation with frozen orthogonally initialized matrices, $k = 256$, and coefficient 1. CLoRA+O-LoRA also evaluates the empty-history overlap term. GORP imports its projection AdamW optimizer, with self-attention LoRA and full-rank cross-attention query/value updates; the latter use learning rate 10^{-4}, projection rank 8, scale 0.25, and refresh interval 10. Historical projection matrices are empty. SPARTA imports its model and adapter-injection modules, retaining rank-2 and rank-8 branches and native routing. It allocates eight routing components for one task, with layer gap 4 and orthogonality weight 10. Its 200-update restoration interval is not reached here. SPARTA+Replay has an empty first-task replay buffer.

InheritLoRA uses GradResp initialization, fixed task reader, static reader-conditioned writer preconditioner, and per-update AdamW-state reset. Reader probes use up to 512 current-task examples (all 250 on CB); static writer statistics use up to 256 examples for T5 and 64 for LLaMA. Only the writer is optimized during task-loss training. We verify trainable parameter changes, unchanged readers, and writer-state reset after every update.

ELLA's trainable parameter counts and timings are measured using a local first-task proxy. The proxy trains ordinary LoRA and constructs a dense adapter update inside the training loop, with no historical penalty in the loss. It omits the full cache and history-management path, including history transfers and storage. ELLA's continual accuracy entries come from the cited sources.

Initialization and complete-task cost. Table 12 separates loading/setup, complete training-loop time, and wall time from the start of model preparation through the end of training. Loading/setup includes model loading, adapter installation, and method-specific preparation. The task total additionally includes data preparation and intervening bookkeeping, but excludes inference. Training-loop wall time includes warm-up, the partial update, synchronization, and per-step logging. For InheritLoRA, loading the separate probe model and constructing the reader together take 26.14/22.01 seconds for T5/LLaMA. Loading the training model and installing the adapter take 1.96/9.17 seconds, followed by 4.01/2.75 seconds for static Gram/preconditioner setup.

Table 12: CB first-task wall times on RTX 5090 (seconds).

Method	Load/setup	Training loop	Task total
T5-large			
SPARTA	5.43	15.85	21.97
SeqLoRA	3.41	6.42	10.48
CLoRA	3.59	7.88	12.09
ELLA	3.04	6.00	9.44
GORP	3.20	18.04	21.79
IncLoRA	3.00	5.85	9.46
O-LoRA	3.20	6.29	10.12
CLoRA + O-LoRA	3.48	8.30	12.40
ASO-LoRA	3.44	6.72	10.59
SPARTA + Replay	4.95	14.68	20.10
Recurrent-KIF	3.15	5.95	9.51
InheritLoRA	32.11	5.72	38.35
LLaMA 3.1–8B			
SPARTA	37.68	54.69	92.80
SeqLoRA	9.24	17.55	27.24
ELLA	9.13	18.82	28.45
O-LoRA	9.19	18.81	28.51
SPARTA + Replay	38.25	55.16	93.97
InheritLoRA	33.93	20.61	55.12

One epoch, seed 42, 250 examples, effective batch 8. Method names refer to the local first-task implementations described above. Load/setup includes the time spent on reader initialization for methods that require this step.

Interpretation. Under the first-task CB protocol with the target-layer and rank settings specified above, InheritLoRA optimizes half as many parameters as O-LoRA on T5 and 38.46% as many on LLaMA. Its measured training-step cost is similar on T5 and higher on LLaMA, while reader initialization increases complete-task cost on this short task. Single-run training timings on four same-model GPUs do not establish a stable ordering for small speed differences.

D Additional Experiments and Diagnostics

The table below maps the experimental questions in § 5 to their supplementary evidence.

Main-text question	Supplementary evidence
Continual learning performance (§ 5.1)	Main results: D.1
Reader selection (§ 5.2)	Reader-only intervention: D.2; matched T5/LLaMA controls: D.3
Writer optimization (§ 5.3)	Default ablation, learning rates, reset frequency, and Gram structure: D.4
Local mechanism and its scope	Filtering: D.5; deficit: D.6; deletion: D.7; protection: D.8; coverage: D.9
Response organization and resource costs	Cross-seed diagnosis: D.10; retained state and computation: D.11

D.1 Main Benchmark Results

Tables 14 and 15 report the fixed-reader InheritLoRA runs underlying Table 2. Each run uses task-start GradResp, Gram-preconditioned writer gradients, and per-update AdamW-state reset. The O4 T5 runs at 10^{-3} are also the GradResp and Full runs in Tables 4 and 5, respectively.

For each run, final accuracy is the equal mean of the task accuracies after the full stream. Each order mean then weights seeds 42–44 equally; standard deviations are sample standard deviations across those seeds. Main-table SC and LS averages weight their three order means equally, using unrounded values. The results cover 36 runs across the two models, six orders, and three seeds. LC and FinalAcc are accuracies (%); BWT and AvgForget are in percentage points. LC, BWT, and AvgForget in Table 13 use the same equal weighting over seeds and task orders; sample standard deviations across seeds are reported separately for each order in Table 15.

Table 13: Task acquisition and retention of InheritLoRA.

Model	Orders	LC ↑	FinalAcc ↑	BWT ↑	AvgForget ↓
T5-large	O1–O3	79.35	78.42	−1.24	1.39
	O4–O6	78.24	75.89	−2.52	2.81
LLaMA 3.1–8B	O1–O3	80.47	79.91	−0.75	0.81
	O4–O6	82.88	82.61	−0.29	0.84

Equal means over seeds 42–44 and the listed orders. Table 15 reports the corresponding sample standard deviations separately for each task order.

Table 14: Seed-level InheritLoRA main results.

Model	Order	Seed 42	Seed 43	Seed 44	Mean ± SD
T5-large	O1	78.52	79.22	77.48	78.41 ± 0.88
	O2	78.87	79.51	77.81	78.73 ± 0.86
	O3	78.66	78.51	77.21	78.13 ± 0.80
	O4	75.55	76.65	76.97	76.39 ± 0.75
	O5	76.42	76.61	76.05	76.36 ± 0.29
	O6	75.47	74.15	75.12	74.91 ± 0.68
LLaMA 3.1–8B	O1	81.10	79.15	77.84	79.36 ± 1.64
	O2	81.24	79.51	79.07	79.94 ± 1.15
	O3	81.29	79.43	80.53	80.42 ± 0.93
	O4	83.80	81.39	82.99	82.73 ± 1.23
	O5	83.27	82.44	83.85	83.19 ± 0.71
	O6	82.62	82.66	80.44	81.90 ± 1.27

Final accuracy (%) after four tasks (O1–O3) or fifteen tasks (O4–O6). The last column gives the three-seed mean ± sample standard deviation.

Table 15: InheritLoRA learning and retention in the main experiments.

Model	Order	LC ↑	FinalAcc ↑	AvgForget ↓	BWT ↑
T5-large	O1	79.28 ± 1.06	78.41 ± 0.88	1.28 ± 0.92	−1.16 ± 1.12
	O2	79.19 ± 1.04	78.73 ± 0.86	0.90 ± 0.64	−0.61 ± 1.15
	O3	79.58 ± 0.75	78.13 ± 0.80	2.00 ± 1.09	−1.94 ± 1.04
	O4	78.14 ± 0.68	76.39 ± 0.75	2.38 ± 0.47	−1.88 ± 0.70
	O5	78.76 ± 0.17	76.36 ± 0.29	2.93 ± 0.25	−2.57 ± 0.16
	O6	77.81 ± 0.31	74.91 ± 0.68	3.13 ± 0.91	−3.10 ± 0.95
LLaMA 3.1–8B	O1	80.19 ± 0.98	79.36 ± 1.64	1.18 ± 1.12	−1.10 ± 1.15
	O2	80.36 ± 0.91	79.94 ± 1.15	0.67 ± 0.43	−0.56 ± 0.39
	O3	80.85 ± 0.85	80.42 ± 0.93	0.58 ± 0.25	−0.58 ± 0.25
	O4	82.68 ± 0.64	82.73 ± 1.23	0.61 ± 0.42	0.05 ± 0.68
	O5	83.36 ± 0.30	83.19 ± 0.71	0.71 ± 0.31	−0.19 ± 0.44
	O6	82.58 ± 1.08	81.90 ± 1.27	1.20 ± 0.32	−0.73 ± 0.59

Three-seed means ± sample standard deviations. LC and FinalAcc are accuracies (%); AvgForget and BWT are in percentage points. Definitions follow Appendix C.3.

D.2 Reader-Only Intervention: Absolute Results

Tab. 16 expands the paired changes in Tab. 1. All arms use T5-large, the complete O4 sequence, one epoch per task, and seeds 42–44. Within each method, only the task-start reader changes.

Table 16: Absolute reader-only intervention results.

Method	Reader	LC ↑	Final ACC ↑	Avg. forget. ↓
O-LoRA	Random	77.68	68.01	10.70
	GradResp	78.02	71.97	6.72
GORP	Random	73.99	67.72	8.71
	GradResp	76.20	73.99	3.37
IncLoRA	Random	77.45	60.89	18.31
	GradResp	75.12	61.32	16.61

Means over three seeds; acquisition and final accuracy are reported as percentages, whereas forgetting measures the accuracy decrease in percentage points.

The complete task-by-stage matrices for these same runs also permit the peak-normalized check in Equation 59. Tab. 17 reports CFr computed within each seed and then aggregated across seeds; all 14 historical peak denominators are strictly positive. The normalized measure decreases under GradResp for all three methods, including IncLoRA ($23.09\% \rightarrow 21.24\%$). For IncLoRA, this decrease accompanies lower LC and a small increase in final accuracy.

Table 17: Reader-only intervention: peak-normalized forgetting for Random and GradResp.

Method	Reader	CFr (%) ↓
O-LoRA	Random	14.38
	GradResp	9.54
GORP	Random	11.34
	GradResp	5.13
IncLoRA	Random	23.09
	GradResp	21.24

CFr is the dimensionless peak-referenced loss fraction in Equation 59, shown as a percentage. Values are means over the same three training seeds used for the absolute results in Tab. 16.

D.3 READER-SELECTION CONTROLS

The matched controls below support Section 5.2. Both backbones use O4, seeds 42–44, rank 8, and one epoch per task. LC and FinalAcc are percentages; AvgForget and BWT are in percentage points. Means weight seeds equally, and $\pm$ denotes sample SD.

D.3.1 T5-LARGE: RANDOM, ACTIVATION-SVD, GRADIENT-SVD, AND GRADRESP

Only the task-start reader construction varies. Each task creates a new reader–writer block with $B_0 = 0$; its reader, the base model, and historical blocks remain frozen while the current writer is optimized. Data, learning-rate schedule, batches, accumulation, clipping, dropout, and evaluation are shared. Gram preconditioning and per-update reset are enabled.

Reader constructions. The random reader uses Kaiming-uniform initialization from a separate reproducible random stream. Activation-SVD takes the leading r right singular vectors of the matched layer-input matrix X; Gradient-SVD takes those of the raw dense-weight loss gradient G. Both use unweighted singular vectors. GradResp uses the response-coordinate ridge construction in Section 4.1 with $\lambda = 10^{-4}$. The shared task-start probe samples up to 512 current-task training examples on the base model without historical adapters, using matched instruction activations and sample-mean loss gradients. Initialization and probing preserve the subsequent training random-number state.

All constructions pass through the same full-matrix sample-standard-deviation scaling and historical filter, including row-norm restoration. The filter uses each run's own previously retained readers. Each installed reader also defines its own Gram preconditioner, estimated from a separate activation sample of the continual model. Gradient-SVD and Activation-SVD both use the common fixed-reader writer protocol.

Table 18 gives every run, and Table 19 gives seed-matched contrasts. BWT summarizes the signed acquisition–retention gap over the 14 earlier tasks, following Equation 58.

Table 18: T5-large/O4 reader controls by seed.

Arm	Seed	LC ↑	FinalAcc ↑	BWT ↑
Random	42	78.66	67.83	−11.61
	43	78.14	67.11	−11.82
	44	78.12	65.45	−13.57
Activation-SVD	42	77.37	76.06	−1.41
	43	77.26	76.12	−1.22
	44	77.98	76.30	−1.80
Gradient-SVD	42	77.54	76.64	−0.96
	43	77.79	77.31	−0.52
	44	78.48	76.73	−1.88
GradResp	42	77.83	75.55	−2.44
	43	77.67	76.65	−1.10
	44	78.91	76.97	−2.08

GradResp reuses the Full writer runs in Table 27.

Table 19: T5-large/O4 paired reader effects.

Contrast	Seed 42	Seed 43	Seed 44	Mean
Activation-SVD − Random	8.23/10.21	9.01/10.60	10.84/11.77	9.36/10.86
Gradient-SVD − Random	8.82/10.65	10.20/11.30	11.28/11.69	10.10/11.22
GradResp − Random	7.72/9.17	9.54/10.72	11.52/11.49	9.59/10.46
Activation-SVD − GradResp	0.51/1.04	−0.53/ − 0.12	−0.67/0.28	−0.23/0.40
Gradient-SVD − GradResp	1.10/1.49	0.66/0.58	−0.24/0.21	0.51/0.76

Each cell reports ΔFinalAcc / ΔBWT in pp; positive values favor the first arm. Means are paired across seeds.

D.3.2 LLaMA 3.1–8B: GradResp versus Random

Protocol and arm definitions. We compare GradResp and Random readers on LLaMA 3.1–8B over the complete 15-task O4 stream, pairing both reader arms at seeds 42–44. Each task trains a new zero-initialized writer while freezing its reader, the backbone, and historical adapters. Table 20 defines the two reader arms. Both use Gram preconditioning and per-update AdamW-state reset.

The archived configuration uses rank 8, LoRA scale 32, dropout 0.1, and one epoch per task, with learning rate 10^{-4}, constant scheduling, no warm-up, and configured weight decay zero. A physical batch of one and 32-step gradient accumulation give effective batch size 32. AdamW uses $(\beta_1, \beta_2) = (0.9, 0.999)$ and $\epsilon = 10^{-8}$; gradients are clipped at norm 1.0. Backbone weights use bfloat16 and writer parameters float32. Source/target limits are 512/50 tokens; evaluation uses batch size 8 and greedy generation with at most 50 new tokens.

Reader construction uses up to 512 current-task training examples with probe length 256, ridge 10^{-4}, and the base model without historical adapters. Random uses an independent random stream, with the same scaling and history filtering. Each run retains its own history and constructs its own reader-induced Gram geometry from up to 64 current-task examples with a 256-token-row budget and spectral cutoff 10^{-2}. Each complete stream has 1,732 optimizer updates and 1,732 resets. The per-task update counts match the archived Trainer handling of incomplete accumulation groups.

Both reader arms are rerun on A800 with PyTorch 2.5.1/CUDA 12.4, Transformers 4.46.3, and Accelerate 1.1.1. The checkpoint hashes and Full hyperparameters match the main O4 experiment; the main-table LLaMA runs use a separate RTX 5090/PyTorch 2.12/CUDA 13.0 implementation snapshot. Every contrast below pairs runs within the A800 cohort. The main-benchmark results retain their independent provenance in Appendix D.1.

Complete results. Tables 20 and 21 report both reader arms. LC and FinalAcc average the 15 task accuracies; AvgForget and BWT cover the 14 earlier tasks, following Equation 57. Means weight the 3 seeds equally; standard deviations are sample SD across seeds.

Table 20: LLaMA reader comparison.

Arm	Reader	Precond.	State	LC ↑	FinalAcc ↑	AvgForget ↓	BWT ↑
L0	GradResp	ON	Reset	82.75 ± 0.26	83.03 ± 0.08	0.47 ± 0.05	0.30 ± 0.20
L1	Random	ON	Reset	80.53 ± 0.29	78.82 ± 0.37	2.46 ± 0.42	−1.83 ± 0.53

O4, seeds 42–44. Entries are mean ± sample SD. LC and FinalAcc are percentages; AvgForget and BWT are in pp. L0 is the full configuration.

Table 21: Individual LLaMA reader runs.

Arm	Seed	LC ↑	FinalAcc ↑	AvgForget ↓	BWT ↑
L0	42	82.5706	83.0098	0.4291	0.4706
L0	43	83.0466	83.1207	0.5172	0.0794
L0	44	82.6360	82.9696	0.4491	0.3574
L1	42	80.6293	79.2251	2.3821	−1.5045
L1	43	80.7664	78.4894	2.9177	−2.4396
L1	44	80.2072	78.7584	2.0818	−1.5523

Each row is one complete 15-task trajectory. Arm definitions are given in Table 20.

Paired contributions and task-level differences. GradResp (L0) achieves higher FinalAcc than Random (L1) in all three paired seeds, with a mean improvement of 4.21 ± 0.42 pp (mean ± sample SD; Table 22). The seed-averaged difference is positive on 12 of the 15 tasks. The largest gains occur on MNLI (20.57 pp) and CB (10.12 pp), while WiC, COPA, and SST-2 decrease by 0.21, 0.67, and 0.08 pp, respectively (Table 26). Per-task results for each seed appear in Tables 23, 24, and 25. The seed is the unit of replication.

Table 22: Paired LLaMA reader contrasts.

Contrast	Seed	ΔLC ↑	ΔFinalAcc ↑	ΔAvgForget ↓	ΔBWT ↑
L0−L1	42	1.9413	3.7847	−1.9530	1.9751
L0−L1	43	2.2802	4.6313	−2.4005	2.5190
L0−L1	44	2.4288	4.2112	−1.6327	1.9097
L0−L1	Mean ± SD	2.22 ± 0.25	4.21 ± 0.42	−2.00 ± 0.39	2.13 ± 0.33

Every entry is the first arm minus the second arm, in percentage points. Positive LC, FinalAcc, and BWT contrasts favor the first arm; negative AvgForget contrasts favor it. Means and sample SD are computed over paired seed differences.

Table 23: LLaMA per-task acquisition and final accuracy, seed 42.

Task	L0 LC	L0 Final	L1 LC	L1 Final
MNLI	80.62	81.68	76.30	60.04
CB	87.50	89.29	78.57	80.36
WiC	59.40	69.12	63.48	67.40
COPA	90.00	90.00	94.00	93.00
QQP	83.18	83.21	82.33	78.07
BoolQA	82.69	82.57	73.36	74.28
RTE	87.00	86.64	83.39	85.92
IMDB	95.67	94.22	89.70	92.61
Yelp	66.16	64.78	65.30	63.87
Amazon	64.54	63.75	63.29	60.71
SST-2	94.50	93.81	95.07	95.30
DBpedia	98.74	98.74	98.82	97.89
AG News	90.86	90.86	90.22	86.70
MultiRC	82.24	81.02	80.88	77.52
Yahoo	75.46	75.46	74.72	74.72

All values are percentages. LC is evaluated immediately after learning the task; Final is evaluated after the complete O4 stream.

Table 24: LLaMA per-task acquisition and final accuracy, seed 43.

Task	L0 LC	L0 Final	L1 LC	L1 Final
MNLI	82.47	83.21	75.34	59.76
CB	89.29	92.86	80.36	78.57
WiC	65.52	61.76	68.65	61.91
COPA	91.00	93.00	92.00	94.00
QQP	82.80	80.42	82.72	77.22
BoolQA	82.91	83.61	71.90	73.43
RTE	87.36	87.00	83.39	83.75
IMDB	95.36	95.80	89.96	92.76
Yelp	67.41	67.05	67.47	63.12
Amazon	62.70	62.42	61.25	58.71
SST-2	94.95	94.95	95.30	94.27
DBpedia	98.84	98.74	98.83	97.88
AG News	89.54	89.86	90.66	89.53
MultiRC	80.53	81.11	78.61	77.37
Yahoo	75.03	75.03	75.05	75.05

All values are percentages. LC is evaluated immediately after learning the task; Final is evaluated after the complete O4 stream.

Table 25: LLaMA per-task acquisition and final accuracy, seed 44.

Task	L0 LC	L0 Final	L1 LC	L1 Final
MNLI	78.71	82.16	75.95	65.53
CB	89.29	87.50	78.57	80.36
WiC	66.46	62.38	66.61	64.58
COPA	92.00	94.00	91.00	92.00
QQP	82.29	83.74	82.49	81.29
BoolQA	78.96	81.25	71.99	73.00
RTE	86.64	86.28	81.95	82.67
IMDB	95.58	96.04	90.16	92.71
Yelp	66.25	66.47	66.72	61.07
Amazon	63.86	64.26	62.38	57.14
SST-2	94.95	95.41	94.50	94.84
DBpedia	98.86	98.79	98.72	98.20
AG News	89.74	90.14	90.14	87.97
MultiRC	80.57	80.71	78.57	76.67
Yahoo	75.39	75.39	73.36	73.36

All values are percentages. LC is evaluated immediately after learning the task; Final is evaluated after the complete O4 stream.

Table 26: Task-level paired changes in final accuracy.

Task	L0−L1
MNLI	20.57
CB	10.12
WiC	−0.21
COPA	−0.67
QQP	3.60
BoolQA	8.91
RTE	2.53
IMDB	2.66
Yelp	3.42
Amazon	4.62
SST-2	−0.08
DBpedia	0.76
AG News	2.22
MultiRC	3.76
Yahoo	0.92

First arm minus second arm, averaged equally over seeds 42–44, in percentage points. The table covers all 15 tasks and includes the GradResp-minus-Random reader contrast.

D.4 Writer-Side Controls

These controls support § 5.3, separating Gram preconditioning, optimizer-state reset, and their operating regimes. The T5-large experiments use O4, seeds 42–44, rank 8, one epoch per task, and the same fixed-reader training protocol as Appendix D.3.1. LC and FinalAcc are percentages; AvgForget and BWT are in percentage points. Means weight the three seeds equally, and $\pm$ denotes sample SD with denominator $3 - 1$. Table 5 reports the LLaMA controls at their default learning rate alongside the corresponding T5 controls.

D.4.1 Default 2×2 Ablation

The default operating rates are 10^{-3} for T5-large and 10^{-4} for LLaMA 3.1–8B. Table 5 reports the four configurations at each backbone's rate; the tables below retain the individual T5 runs (Table 27) and paired component effects (Table 28).

The 2×2 comparison uses GradResp throughout. The OFF condition omits the Gram transformation of writer gradients while retaining the same activation collection and Gram construction as the ON condition. Reset follows § 4.2 after each accumulated optimizer update. The reset runs contain 866

optimizer updates and 866 resets per complete stream. BWT summarizes the signed acquisition–retention gap over the 14 earlier tasks, following Equation 58.

Table 27: T5-large/O4 default writer controls by seed.

Arm	Seed	LC ↑	FinalAcc ↑	BWT ↑
Full: ON, Reset	42	77.83	75.55	−2.44
	43	77.67	76.65	−1.10
	44	78.91	76.97	−2.08
OFF, Preserve	42	77.36	73.86	−3.75
	43	78.53	71.31	−7.74
	44	78.34	72.51	−6.24
OFF, Reset	42	77.99	76.24	−1.87
	43	77.51	73.84	−3.93
	44	77.82	76.24	−1.70
ON, Preserve	42	78.57	75.82	−2.95
	43	77.19	73.76	−3.68
	44	78.18	76.05	−2.28

Full uses preconditioning and reset; Preserve retains AdamW state within each task.

Table 28: T5-large/O4 paired writer-component effects.

Contrast	Seed 42	Seed 43	Seed 44	Mean
Preconditioning, Preserve	1.96/0.80	2.45/4.06	3.54/3.96	2.65/2.94
Preconditioning, Reset	−0.70/ − 0.57	2.81/2.83	0.73/ − 0.38	0.95/0.63
Reset, Preconditioning OFF	2.39/1.88	2.53/3.80	3.73/4.54	2.88/3.41
Reset, Preconditioning ON	−0.27/0.51	2.89/2.58	0.92/0.20	1.18/1.10

Each cell reports ΔFinalAcc / ΔBWT in pp; positive values favor enabling the named factor. Means are paired across seeds.

D.4.2 Learning-Rate Sweep

Cross-backbone control grid. Table 29 reports the backbone-by-rate control grid. The main operating-rate subset appears in Table 5.

T5 sweep design. The T5 sensitivity design comprises 57 unique complete O4 trajectories across learning rate, reset frequency, and Gram structure. Of these, 39 cover the three interior learning rates and the reset-frequency and Gram-structure controls: 12 are shared with the default 2×2 comparison, and 27 cover the other conditions. The remaining 18 cover the two learning-rate endpoints. Each condition uses seeds 42–44 and the same 15-task training and evaluation protocol. LC and FinalAcc are percentages; AvgForget and BWT are in percentage points. Means weight the three seeds equally, and $\pm$ denotes sample standard deviation. AvgForget averages the positive acquisition-to-final accuracy drops over the first 14 tasks; BWT averages their signed changes.

The learning-rate grid is $\{10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}, 10^{-2}\}$ for Full (complete Gram and per-update reset), NoPrecond (identity preconditioning and per-update reset), and NoReset (complete Gram and task-local state preservation). Each configuration uses a constant schedule, rank 8, one epoch per task, physical batch size 16, gradient accumulation 4, zero weight decay, and clipping threshold 1. Each run retains its own historical readers and constructs its own Gram matrices. The grid reports sensitivity under the common protocol; the main comparison retains the learning rate specified in its original experimental protocol.

Across the tested learning rates from 10^{-4} to 3×10^{-3}, Full achieves higher mean FinalAcc than NoReset (Table 30). The mean Full-minus-NoPrecond difference ranges from -2.13 to 3.87 pp (Table 31). At 10^{-4}, Full has lower FinalAcc than NoPrecond (74.05% versus 76.18%), along with lower LC and AvgForget. At 3×10^{-4}, Full and NoPrecond have similar FinalAcc (76.78% and 76.74%), with different acquisition and retention profiles: Full has lower LC (77.10% versus 77.60%) and lower AvgForget (1.22 versus 1.88 points). At 3×10^{-3}, both components contribute more relative to their removal controls, while all three configurations have lower final accuracy

Table 29: Writer controls across backbones and learning rates.

Rate	Precond.	AdamW state	T5-large			LLaMA 3.1–8B		
			LC ↑	FinalAcc ↑	AvgForget ↓	LC ↑	FinalAcc ↑	AvgForget ↓
10^{-3}	OFF	Preserve	78.07 ± 0.63	72.56 ± 1.27	6.27 ± 2.03	46.03 ± 0.07	3.39 ± 0.25	45.69 ± 0.34
	OFF	Reset	77.77 ± 0.24	75.44 ± 1.39	3.13 ± 1.15	46.77 ± 2.90	3.33 ± 1.33	46.54 ± 1.77
	ON	Preserve	77.98 ± 0.71	75.21 ± 1.26	3.30 ± 0.86	43.59 ±13.33	3.60 ± 2.28	42.85 ±11.84
	ON	Reset	78.14 ± 0.68	76.39 ± 0.75	2.38 ± 0.47	49.87 ± 2.66	10.30 ± 6.27	42.40 ± 3.93
10^{-4}	OFF	Preserve	75.28 ± 0.61	76.24 ± 0.31	1.45 ± 0.09	83.51 ± 0.05	82.89 ± 0.45	1.06 ± 0.26
	OFF	Reset	76.24 ± 0.17	76.18 ± 0.45	1.60 ± 0.25	82.43 ± 0.95	82.17 ± 1.14	0.88 ± 0.27
	ON	Preserve	70.38 ± 1.70	70.04 ± 0.82	1.78 ± 0.60	82.61 ± 0.44	82.74 ± 0.44	0.64 ± 0.42
	ON	Reset	73.42 ± 0.80	74.05 ± 0.69	1.42 ± 0.42	82.75 ± 0.26	83.03 ± 0.08	0.47 ± 0.05

O4, seeds 42–44, rank 8, one epoch per task; GradResp readers fixed within tasks. Values are means ± sample SD; LC and FinalAcc are percentages, AvgForget is in pp.

Table 30: T5-large/O4 learning-rate sweep.

Rate	Configuration	LC ↑	FinalAcc ↑	AvgForget ↓	BWT ↑
10^{-4}	Full	73.42 ± 0.80	74.05 ± 0.69	1.42 ± 0.42	0.68 ± 1.46
	NoPrecond	76.24 ± 0.17	76.18 ± 0.45	1.60 ± 0.25	−0.05 ± 0.60
	NoReset	70.38 ± 1.70	70.04 ± 0.82	1.78 ± 0.60	−0.36 ± 1.27
3×10^{-4}	Full	77.10 ± 0.46	76.78 ± 0.37	1.22 ± 0.14	−0.34 ± 0.15
	NoPrecond	77.60 ± 0.15	76.74 ± 0.32	1.88 ± 0.30	−0.92 ± 0.38
	NoReset	76.67 ± 0.38	75.81 ± 0.36	1.95 ± 0.23	−0.93 ± 0.54
10^{-3}	Full	78.14 ± 0.68	76.39 ± 0.75	2.38 ± 0.47	−1.88 ± 0.70
	NoPrecond	77.77 ± 0.24	75.44 ± 1.39	3.13 ± 1.15	−2.50 ± 1.24
	NoReset	77.98 ± 0.71	75.21 ± 1.26	3.30 ± 0.86	−2.97 ± 0.70
3×10^{-3}	Full	76.33 ± 0.86	72.31 ± 1.32	4.57 ± 1.03	−4.31 ± 1.14
	NoPrecond	74.92 ± 0.80	68.43 ± 2.08	7.39 ± 1.41	−6.96 ± 1.79
	NoReset	75.51 ± 0.33	70.14 ± 1.59	6.05 ± 1.55	−5.75 ± 1.54
10^{-2}	Full	48.43 ± 9.02	2.81 ± 1.49	48.88 ± 8.11	−48.88 ± 8.11
	NoPrecond	40.26 ± 2.11	1.20 ± 0.27	41.85 ± 2.53	−41.85 ± 2.53
	NoReset	36.77 ± 3.77	7.85 ± 11.88	30.98 ± 8.79	−30.98 ± 8.79

Complete three-seed conditions.

Table 31: Paired final-accuracy changes across learning rates.

Rate	Contrast	Seed 42	Seed 43	Seed 44	Mean
10^{-4}	Full − NoPrecond	−1.98	−2.74	−1.68	−2.13
	Full − NoReset	+3.80	+3.40	+4.84	+4.01
3×10^{-4}	Full − NoPrecond	+0.00	−0.38	+0.51	+0.04
	Full − NoReset	+0.74	+0.52	+1.68	+0.98
10^{-3}	Full − NoPrecond	−0.70	+2.81	+0.73	+0.95
	Full − NoReset	−0.27	+2.89	+0.92	+1.18
3×10^{-3}	Full − NoPrecond	+5.41	+4.02	+2.19	+3.87
	Full − NoReset	+4.80	+0.20	+1.48	+2.16
10^{-2}	Full − NoPrecond	+1.98	+2.98	−0.12	+1.61
	Full − NoReset	+2.17	−17.67	+0.39	−5.04

Differences are in percentage points and are computed before rounding. Reported means give equal weight to each of the three matched training-seed pairs.

than at 3×10^{-4} and 10^{-3}. At 10^{-2}, mean FinalAcc falls to 2.81%, 1.20%, and 7.85% for Full, NoPrecond, and NoReset, respectively. Full exceeds NoPrecond by 1.61 points but is 5.04 points below NoReset; the latter contrast reverses sign across seeds (Tables 31 and 32).

Table 32: T5-large/O4 endpoint trajectories by seed.

Rate	Configuration	Seed	LC ↑	FinalAcc ↑	AvgForget ↓	BWT ↑
10^{-4}	Full	42	73.32	74.71	1.05	1.48
		43	74.26	73.33	1.87	−1.00
		44	72.66	74.12	1.35	1.56
	NoPrecond	42	76.09	76.68	1.36	0.64
		43	76.42	76.07	1.86	−0.38
		44	76.20	75.80	1.57	−0.42
	NoReset	42	71.29	70.91	1.15	−0.41
		43	71.43	69.93	2.34	−1.61
		44	68.41	69.28	1.85	0.93
10^{-2}	Full	42	54.08	3.43	54.26	−54.26
		43	53.19	3.89	52.82	−52.82
		44	38.03	1.12	39.55	−39.55
	NoPrecond	42	37.88	1.45	39.04	−39.04
		43	41.92	0.92	43.94	−43.94
		44	40.99	1.24	42.58	−42.58
	NoReset	42	33.99	1.26	35.06	−35.06
		43	41.06	21.56	20.89	−20.89
		44	35.25	0.73	36.99	−36.99

Each row reports one complete 15-task trajectory.

D.4.3 Reset Frequency

With complete Gram preconditioning and learning rate 10^{-3}, we compare resetting every 1, 4, or 16 optimizer updates with preserving state within each task. A reset clears the current writer's step counter and both moment buffers after the designated optimizer update, while preserving writer weights and the learning-rate schedule. Each task creates a new writer and optimizer state, and restarts the reset counter. Thus the number of resets at interval k is the sum of $\lfloor n_t/k \rfloor$ over tasks, where n_t is the number of optimizer updates in task t. Every trajectory contains 866 optimizer updates; Every-1, Every-4, and Every-16 execute 866, 208, and 42 resets, respectively, for each seed. Preserve keeps optimizer state throughout each task.

Table 33: T5-large/O4 reset-frequency controls.

Frequency	LC ↑	FinalAcc ↑	AvgForget ↓	BWT ↑
Every-1	78.14 ± 0.68	76.39 ± 0.75	2.38 ± 0.47	−1.88 ± 0.70
Every-4	79.03 ± 0.38	76.19 ± 0.77	3.21 ± 0.54	−3.05 ± 0.60
Every-16	77.02 ± 0.81	74.74 ± 0.47	2.72 ± 1.05	−2.44 ± 1.31
Preserve	77.98 ± 0.71	75.21 ± 1.26	3.30 ± 0.86	−2.97 ± 0.70

Means and sample standard deviations over seeds 42–44; every reset-frequency condition uses complete Gram preconditioning for its writer updates.

Every-4 is 0.20 points below Every-1 in mean FinalAcc, with higher LC (79.03% versus 78.14%) and greater AvgForget (3.21 versus 2.38 points; Table 33). The paired Every-1 minus Every-4 FinalAcc differences are -1.48, $+0.62$, and $+1.46$ points for seeds 42, 43, and 44 (Table 34). Every-16 reaches 74.74%, below Preserve at 75.21%, so performance varies non-monotonically with the interval. These results identify frequent reset settings with similar mean final performance and distinct acquisition–retention trade-offs.

D.4.4 Gram Structure

We fix per-update reset and learning rate 10^{-3}, and vary the matrix right-multiplying the writer gradient. Full-Gram uses the complete preconditioner H obtained by the method's truncated pseudoinverse; Diagonal first constructs that same H and then uses $\mathrm{diag}(\mathrm{diag}(H))$; Identity uses the

Table 34: Individual seeds for the reset-frequency comparison.

Frequency	Seed	LC ↑	FinalAcc ↑	AvgForget ↓	BWT ↑
Every-1	42	77.83	75.55	2.81	−2.44
	43	77.67	76.65	1.88	−1.10
	44	78.91	76.97	2.44	−2.08
Every-4	42	79.44	77.03	2.77	−2.58
	43	78.69	76.03	3.05	−2.84
	44	78.98	75.51	3.82	−3.72
Every-16	42	76.51	74.71	2.23	−1.92
	43	76.60	75.23	2.01	−1.47
	44	77.95	74.28	3.94	−3.93
Preserve	42	78.57	75.82	3.27	−2.95
	43	77.19	73.76	4.18	−3.68
	44	78.18	76.05	2.45	−2.28

identity matrix. Thus Diagonal retains reader-derived coordinate scaling, with the same spectral truncation applied before diagonal extraction. All three arms use the same activation-collection protocol and construct their Gram matrices from the collected activations.

Table 35: T5-large/O4 Gram-structure controls.

Structure	LC ↑	FinalAcc ↑	AvgForget ↓	BWT ↑
Full-Gram	78.14 ± 0.68	76.39 ± 0.75	2.38 ± 0.47	−1.88 ± 0.70
Diagonal	78.34 ± 0.41	76.49 ± 0.50	2.51 ± 0.73	−1.99 ± 0.52
Identity	77.77 ± 0.24	75.44 ± 1.39	3.13 ± 1.15	−2.50 ± 1.24

Means and sample standard deviations over seeds 42–44; every arm uses per-update reset.

Table 36: Paired final-accuracy changes across Gram structures.

Contrast	Seed 42	Seed 43	Seed 44	Mean
Full-Gram − Diagonal	−0.75	−0.41	+0.86	−0.10
Full-Gram − Identity	−0.70	+2.81	+0.73	+0.95
Diagonal − Identity	+0.05	+3.21	−0.13	+1.05

Differences are in percentage points and are computed before rounding.

Diagonal and Full-Gram achieve similar mean FinalAcc (76.49% and 76.39%). Diagonal has slightly higher LC and AvgForget, while both preconditioned variants exceed Identity in mean final accuracy (Table 35). All three pairwise FinalAcc differences change sign across seeds (Table 36). Table 37 shows the task-level acquisition and final-accuracy contrasts, whose signs vary across individual tasks in the sequence.

D.4.5 Fixed-State One-Step Response Fitting

Question and protocol. Does the one-step prediction of Corollary 3.4 hold for response matrices extracted from trained models? We reuse the matched gradient–activation probes of Appendix D.6 at 12 actual T5-large/O4 task-end states: MNLI, CB, RTE, and DBpedia, with seeds 42–44. Checkpoint hashes match the archived probes. All 144 adapted projections are included at each state, giving 1,728 layer–state checks. For each layer, we fix its current rank-8 reader A, absorb the LoRA scale $\alpha/r = 4$ into the checkpoint writer B, and set $Z = AX^\top$ and $R = -GX^\top$ ($\eta_{\text{loc}} = 1$). Every arm uses the same B, R, Z and minimizes the same frozen quadratic $f_A(B) = \frac{1}{2}\|R - BZ\|_F^2$; no training is performed. The cached probes use up to 64 stratified development examples (27 for CB), with at most 32 activation rows per example.

Updates and numerical reference. We compare Identity, $H = I$; Diagonal, $H = \operatorname{diag}(\operatorname{diag}(K_Z^\dagger))$; and Full, $H = K_Z^\dagger$, where $K_Z = ZZ^\top$. Each arm uses $B^+ = B - \eta \nabla_B f_A(B) H$.

Table 37: Task-level effects of Gram structure.

Task	Full-Gram − Diagonal		Full-Gram − Identity	
	Acquisition	Final	Acquisition	Final
MNLI	+0.24	+2.62	+1.53	+6.90
CB	+0.60	−1.79	+2.38	+7.14
WiC	−4.70	+1.46	+3.71	−2.46
COPA	+0.00	−1.00	−3.33	−3.33
QQP	−0.59	−0.95	−0.31	−0.45
BoolQA	+2.00	+1.26	+1.72	+0.10
RTE	+2.17	+1.81	+0.24	+1.93
IMDB	−1.22	−0.45	−0.90	−0.07
Yelp	−1.75	−3.19	−1.73	+0.96
Amazon	−0.29	−2.85	+0.72	+1.62
SST-2	−0.19	+0.00	+0.19	−0.34
DBpedia	+0.24	+0.30	+0.01	+0.09
AG News	−0.25	+0.29	+0.21	+0.40
MultiRC	+0.33	+0.65	+0.43	+1.14
Yahoo	+0.35	+0.35	+0.58	+0.58

Each entry is the mean paired accuracy difference over seeds 42–44, in percentage points. Acquisition is measured immediately after learning the task; Final is measured after the full O4 stream.

Alongside $\eta = 1$, we compute the exact optimal scalar step along each direction, separating its geometry from the choice of step size. The oracle fitted response is RP_Z, and the writer gap is measured directly as $g(B) = \frac{1}{2}\|BZ - RP_Z\|_F^2$. All calculations use float64 after promoting cached float16 activations and float32 gradients. Full applies the equivalent factored correction $(R-BZ)Z^\dagger$, avoiding the squared condition number of an explicit Gram solve. Numerical rank uses the singular-value threshold $\max(\mathrm{shape}(Z))\epsilon_{64}\sigma_{\max}(Z)$; all 24 rank-deficient cases are retained. An independent LAPACK least-squares solve checks 48 cases selected by layer type, including the three most ill-conditioned CB cross-query layers.

Table 38: Remaining writer gap after one fixed-state update.

Conditioning	Unit step	Optimal scalar step
Full, exact	5.206×10^{-30}	4.703×10^{-30}
Diagonal, exact inverse	1.371×10^{22}	0.3474
Identity	1.610×10^{5}	0.2711
Full, 10^{-2} cutoff	0.01193	0.01193
Diagonal, 10^{-2} cutoff	56.11	0.3013

Values are $g(B^+)/g(B)$, averaged equally over layers within each state and then over the 12 states. Layers are not independent statistical replicates. Cutoff controls apply the implementation's relative spectral threshold to this same matched-probe Gram matrix.

Results and scope. Full reaches the oracle fit in all 1,728 checks: its largest remaining gap ratio is 3.10×10^{-28}, and the independent solver's largest ratio is 2.37×10^{-28}. Even with optimal scalar steps, Identity and Diagonal retain mean gap ratios of 0.2711 and 0.3474, respectively (Table 38). The exact inverse can have large diagonal entries in ill-conditioned coordinates; optimal scalar steps account for the resulting step-size sensitivity of Diagonal. Applying the 10^{-2} eigenvalue cutoff leaves a mean Full gap ratio of 0.01193, with a maximum of 0.8724, so spectral truncation changes the one-step fit. This numerical check validates the exact local least-squares statement. The CE+AdamW controls in Appendix D.4.4, which use task-start geometry, separately show that diagonal conditioning can attain final accuracy comparable to the full preconditioner.

D.5 Historical Filtering and Response Coverage

Historical filtering reduces overlap with previously retained reader directions, which can also remove current-response coordinates. We measure this effect for CB, the second task in O4, at seed 42

across all 144 adapted layers. On one matched base-model probe, we construct Gradient-SVD and GradResp readers and evaluate them before filtering and after the actual scale–filter installation. Each arm uses its own MNLI history. With the energy-normalized deficit $2\mathcal{D}_{R,X}(A)/\|R\|_F^2$, the layer-mean value rises from 0.0446% to 34.44% for Gradient-SVD and from $8.6\times10^{-6}\%$ to 28.35% for GradResp. These increases quantify the current-response coverage cost of filtering the task-informed readers against their retained histories.

D.6 Task-End Deficit and Long-Horizon Outcomes

Across 168 task-end states, reader deficit does not consistently order subsequent forgetting. We examine its associations with next-task damage and stream-end forgetting using covariate adjustment and reader comparisons matched by task and seed.

Acquisition at individual tasks. We supplement the 12 complete T5-large/O4 trajectories in Table 4 with acquisition accuracy and task-specific test NLL at all 180 task endpoints (Table 39). Seeds are weighted equally. For Activation-SVD, Gradient-SVD, and GradResp, respectively, the median absolute acquisition-accuracy difference from Random within the same task and seed is 1.32, 1.26, and 1.10 pp; the maxima are 14, 6, and 16 pp. Thus similar average LC coexists with differences at individual tasks. Each acquisition NLL, including Yahoo, uses that task's complete test set; the diagnostics below use fixed evaluation samples.

Table 39: Task-specific acquisition under the common fixed-reader protocol.

	Random		Activation-SVD		Gradient-SVD		GradResp	
Task	Acc.	NLL	Acc.	NLL	Acc.	NLL	Acc.	NLL
MNLI	82.63	0.3783	84.17	0.4143	84.75	0.4096	85.09	0.4574
CB	89.29	0.0877	89.29	0.0785	91.67	0.0643	91.67	0.0604
WiC	54.65	0.2452	54.86	0.2456	51.15	0.2761	54.28	0.2334
COPA	59.67	0.3627	51.33	0.4274	55.00	0.4244	51.67	0.4555
QQP	85.75	0.1067	83.57	0.1181	84.57	0.1134	85.14	0.1074
BoolQA	81.78	0.1669	81.30	0.1668	81.30	0.1656	81.49	0.1651
RTE	85.92	0.1139	83.75	0.1248	85.32	0.1159	86.28	0.1104
IMDB	93.70	0.0885	93.88	0.0891	94.37	0.086	93.16	0.1008
Yelp	61.76	0.3598	61.55	0.3872	58.97	0.3962	58.60	0.3964
Amazon	57.77	0.4047	59.22	0.4065	58.57	0.402	59.61	0.4011
SST-2	93.73	0.1087	93.96	0.1049	94.15	0.1079	94.00	0.1104
DBpedia	98.69	0.0371	98.59	0.0444	98.52	0.0432	98.59	0.0436
AG News	87.82	0.1707	88.78	0.1701	88.82	0.1568	88.78	0.1696
MultiRC	70.44	0.1798	72.08	0.1746	71.30	0.1741	71.18	0.1755
Yahoo	71.05	0.2143	72.76	0.1935	72.35	0.1982	72.54	0.2046

Means over seeds 42–44 at the endpoint of each task. Accuracy is in percent; NLL is evaluated on that task's complete test set. Task order follows O4. NLL scales differ across tasks.

States, measurements, and endpoints. We diagnose the first 14 task endpoints in each trajectory, giving $4 \times 3 \times 14 = 168$ states. At each actual accumulated model state, matched activations and gradients define the response reference and the deficit of the retained reader. The primary metric is

$$\bar{d}_t = \frac{1}{144}\sum_{\ell=1}^{144} \frac{2\mathcal{D}_{R_{t,\ell},X_{t,\ell}}(A_{t,\ell})}{\|R_{t,\ell}\|_F^2}. \tag{60}$$

The rank floor in Equation 5 is subtracted before normalization. Energy-weighted aggregation is a secondary sensitivity analysis. Normalization uses each measured positive response energy without an epsilon floor. Of 24,192 layer-state observations, 2,991 fall below 10^{-12} of their state's maximum response energy and remain in the primary mean; this motivates the energy-weighted check. Diagnostic gradient and loss-evaluation sets are disjoint: 64 examples each, or 27 each for CB; the same sample identities are used across readers and seeds. These old-task samples form the offline diagnostic set. We separately measure next-task damage and stream-end forgetting, retaining both signed accuracy drops and their positive part. Both next-task and stream-end accuracies use complete test sets. Next-task accuracy compares two FP32 endpoints with TF32 disabled; 36 CPU cross-checks agree on all 293,870 predictions.

Association and comparison structure. Table 40 reports the primary metric. Adjustment residualizes task and seed fixed effects, acquisition accuracy, and acquisition diagnostic NLL; task effects also absorb sequence position. The associations vary with covariate adjustment and reader family. We also report correlations within the task-informed reader group to characterize this variation. Table 41 uses all 126 same-task, same-seed contrasts against Random. These contrasts share their Random control, and all states originate from 12 trajectories. Crossed task/seed block resampling preserves these shared controls and reader comparisons. With 1,000 resamples, the paired primary Pearson percentile intervals are $[-0.172, 0.505]$ for next-task NLL damage and $[-0.475, 0.707]$ for final clipped forgetting. These intervals summarize uncertainty across the three training seeds.

Table 40: Associations of the primary task-end deficit with subsequent damage.

Reader set	Outcome	r	ρ	Adj. r	Adj. ρ
All four	Next-task NLL change	0.136	0.137	0.223	0.195
	Next-task accuracy drop	0.079	0.131	0.174	0.248
	Final clipped forgetting	0.020	0.073	0.455	0.270
	Final signed accuracy drop	0.029	0.063	0.448	0.262
	Final NLL change	−0.212	−0.188	0.355	0.149
Task-informed only	Next-task NLL change	0.127	0.198	−0.086	0.095
	Next-task accuracy drop	0.040	0.107	−0.092	0.158
	Final clipped forgetting	−0.453	−0.140	−0.251	−0.154
	Final signed accuracy drop	−0.404	−0.154	−0.273	−0.163
	Final NLL change	−0.471	−0.303	−0.129	−0.035

r: Pearson; ρ: Spearman. All four readers contribute 168 states; task-informed readers contribute 126. All outcomes concern the old task; positive changes denote damage. Adjusted correlations control the covariates specified in the text.

Table 41: Deficit and outcome contrasts relative to Random.

	Layer mean (primary)		Energy weighted	
Outcome contrast	r	ρ	r	ρ
Next-task NLL change	0.137	0.189	0.112	0.152
Next-task accuracy drop	0.106	0.112	0.073	0.037
Final clipped forgetting	0.155	0.065	0.034	0.021
Final signed accuracy drop	0.150	0.065	0.038	0.017
Final NLL change	0.281	0.251	0.185	0.179

All 126 paired contrasts are included: three task-informed readers, 14 tasks, and three seeds. Both deficit and outcome differences subtract the same-task, same-seed Random value.

Stability and protection residuals. Across 72 fixed-pool probe resamples on CB and RTE, the absolute change in the primary deficit has median 0.336 pp and maximum 1.692 pp; corresponding energy-weighted changes are 1.388 and 6.626 pp. These resamples measure probe sensitivity within the fixed pool and trained trajectories. All 168 measured deficits are positive; subsequent diagnostic NLL rises in 80 states, including 52 with positive first-order harm from the unprotected update component. These counts describe realized damage within the positive-deficit cohort.

We measure historical protection by the relative Frobenius norm $\|\Delta W P_H\|_F / \|\Delta W\|_F$, aggregated over adapted layers. Here ΔW is the actual effective checkpoint increment, including the LoRA forward scale, and P_H is the implemented joint historical-reader projector. We also separate the protected and unprotected contributions to the first-order old-task loss change.

D.7 Deleting Unprotected Sensitivity from Actual Updates

We examine the unprotected sensitivity described in Corollary 3.3 by deleting old-task-sensitive components from actual checkpoint updates and evaluating how these deletions change the losses on both the old task and the new task.

Intervention and controls. We use Random and GradResp, seeds 42–44, and four transitions: MNLI→CB, CB→WiC, RTE→IMDB, and DBpedia→AG News. For each of the 24 cases, the

two real checkpoints define the effective weight increment ΔW, including the LoRA forward scale. With P_H the implemented joint historical-reader projector, we obtain old-task sensitivity outside this boundary as $G_{\text{old}}(I - P_H)$. Targeted deletion removes update components along its leading input directions: two per layer, or one if the numerical rank is insufficient. This gives 3,409 layer cases with two removed directions and 47 with one removed direction.

Each case includes 20 random deletions in the allowed space, matched to the deleted dimension and the targeted arm's remaining update norm, plus whole-update norm-matched shrinkage. We retain the original random arm as well as the unmodified and targeted updates, totaling 576 evaluated arms. Direction estimation and loss evaluation use disjoint samples. The old- and new-task losses are evaluated before the original update, after that update, and after each counterfactual intervention. This is an offline diagnostic using old-task gradients.

Effects and costs. Targeted deletion improves old-task NLL and outperforms every one of the 20 random controls and norm-matched shrinkage in 20/24 cases. Its median old-task improvement is 0.0059 NLL. New-task NLL worsens relative to the unmodified update in 19/24 cases (median cost 0.009615), while all 24 retain new-task improvement over the pre-update model. In 15/24, old-task NLL remains above its pre-update value. The reported medians summarize changes on the native NLL scale of each task, without normalization across tasks.

All 19 cases whose original update increases old-task NLL improve under targeted deletion; only one of the five originally non-harmful cases does. Table 42 gives a post-hoc breakdown of the four cases in which deletion increases old-task loss. The maximum relative remaining-norm mismatch is 1.51×10^{-9}; the maximum protection residual among these 24 actual updates is 1.43×10^{-6}. The intervention therefore isolates a contribution of unprotected sensitivity to realized interference, together with its new-task cost.

Table 42: Cases in which targeted deletion increases old-task loss.

Reader	Seed	Old task	Before	Original	Targeted	New cost
GradResp	42	DBpedia	0.001393	0.001047	0.00118	+0.009935
GradResp	43	RTE	0.1017	0.1001	0.1046	−0.000228
GradResp	44	DBpedia	0.004597	0.000162	0.000238	−0.001696
Random	42	RTE	0.123	0.09972	0.1227	+0.01815

Before, Original, and Targeted are old-task NLL values. New cost is the targeted minus original new-task NLL. All four original updates improve old-task loss; deleting sensitive components can remove beneficial changes.

Paired reader differences and deletion strength. We pair Random and GradResp by transition and seed, retaining all 12 pairs on identical evaluation examples. Each reader's old-task damage is $D_j(\alpha) = L_{j,\text{old}}(\alpha) - L_{j,\text{old}}^{\text{before}}$, where j denotes the reader and α the deletion strength. We compare their damage using $G(\alpha) = D_{\text{Random}}(\alpha) - D_{\text{GradResp}}(\alpha)$, and distinguish the signed reduction $C(\alpha) = G(0) - G(\alpha)$ from the absolute gap closure $A(\alpha) = |G(0)| - |G(\alpha)|$. Positive A means that the two damage values become closer; positive C means that Random's damage decreases relative to GradResp's. These differ when $G(0) < 0$.

We evaluate $\Delta W_\alpha = \Delta W - \alpha \Delta W_{\text{removed}}$ at $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$ using the same 24 checkpoint cases. Here $\Delta W_{\text{removed}}$ is the component deleted in the targeted intervention above. At every nonzero strength, the 20 random controls match deletion dimension and the targeted arm's remaining per-layer update norm; whole-update shrinkage matches the same norm. Endpoint checks reproduce the original and full-deletion evaluations. Old-task direction estimation and evaluation remain disjoint. New-task calibration uses a separate class-stratified development half; the original held-out half supplies the reported new-task costs.

Random has greater original damage in 9/12 pairs. Full deletion closes the absolute gap in all nine and in 11/12 pairs overall (Table 43); the signed gap decreases in 10/12. The exception in absolute closure is MNLI→CB, seed 43, where GradResp originally has greater damage and the gap grows. For each nonzero strength, the number of pairs with positive absolute closure equals the number exceeding both shrinkage and every one of the 20 paired random controls (Table 44). Random controls are paired by repetition index across readers. Their repetitions and the five strengths are

repeated diagnostics of the same training trajectories. All NLL differences remain on each task's native loss scale, with no rescaling across tasks.

Table 43: All paired reader differences under full targeted deletion.

Window	Seed	$G(0)$	$G(1)$	$A(1)$	Cost: Random	Cost: GradResp
MNLI→CB	42	+0.01063	−0.002686	+0.007944	+0.003988	−0.00216
MNLI→CB	43	−0.009364	−0.01093	−0.001567	+0.01521	+0.007096
MNLI→CB	44	+0.03067	+0.02134	+0.009332	+0.01224	+0.01005
CB→WiC	42	+0.04089	−0.001168	+0.03973	+0.04482	−0.004328
CB→WiC	43	+0.02321	+0.0116	+0.0116	+0.03318	+0.002155
CB→WiC	44	−0.1667	−0.08783	+0.0789	+0.0537	+0.000106
RTE→IMDB	42	−0.02807	−0.004389	+0.02368	+0.01815	+0.00353
RTE→IMDB	43	+0.0273	−0.000934	+0.02637	+0.01622	−0.000228
RTE→IMDB	44	+0.185	+0.004274	+0.1807	+0.009296	+0.004274
DBpedia→AG News	42	+0.000497	+0.000217	+0.00028	+0.03	+0.009935
DBpedia→AG News	43	+0.000101	+0.000015	+0.000086	+0.03605	−0.005529
DBpedia→AG News	44	+0.004582	+0.00435	+0.000232	+0.03816	−0.001696

All values are NLL differences. G compares each reader's change from its own pre-update loss; $A > 0$ denotes absolute gap closure. New-task cost is held-out NLL after deletion minus NLL after the original update; a negative cost denotes improvement.

Table 44: Paired outcomes across the deletion-strength grid.

α	$A > 0$	$C > 0$	$A > A_{\text{random}}$	$A > A_{\text{shrink}}$	$\text{Cost}_{\text{R}} > \text{Cost}_{\text{G}}$
0.25	10/12	11/12	10/12	10/12	8/12
0.50	10/12	11/12	10/12	10/12	9/12
0.75	11/12	10/12	11/12	11/12	11/12
1.00	11/12	10/12	11/12	11/12	12/12

Counts use all 12 transition–seed pairs at each strength. Exceeding random controls means exceeding every one of 20 paired controls. R and G denote Random and GradResp; costs are evaluated as changes in new-task NLL on held-out examples.

Acquisition cost and common-budget comparisons. Full deletion incurs greater held-out new-task NLL cost for Random in 12/12 pairs. To examine this imbalance, we select strengths separately for the two readers using only new-task calibration costs. For each pair, let $b_{\max}$ be the smaller of the two readers' maximum positive targeted costs on the five-point grid. We use budgets $\{0.25, 0.5, 0.75, 1\}b_{\max}$ and select the measured strength with calibration cost closest to each budget from below, breaking ties toward the smaller strength. We apply the same selection rule separately to targeted deletion, shrinkage, and each random control. All selected points are retained in the analysis, irrespective of the direction or magnitude of their held-out effects.

Seven of the 12 pairs have positive common support, yielding 28 targeted budget points; all five unsupported pairs remain in the fixed-strength analysis. Eighteen of these 28 points close the absolute damage gap. The selected costs differ between readers by more than $0.1b_{\max}$ at 22/28 calibration points and 20/28 held-out points. At 15/28 points, at least one reader selects zero deletion. One point combines nonzero deletion for both readers with cost differences within this tolerance on both splits. The 28 points are repeated measurements within seven transition–seed pairs.

Thus, the strength sweep establishes local gap closure across several intervention strengths together with unequal acquisition costs. The common-budget grid provides one nonzero point with matched realized costs on both splits, leaving insufficient coverage to estimate the effect across task transitions at equal acquisition cost.

D.8 Intervening on the Inherited Protection State

We isolate the effect of inherited protection by exchanging only the retained-reader copy used for future historical projection, while holding the learned parent parameters and child data order fixed.

Construction. On T5-large/O4, we prespecify CB→WiC, RTE→IMDB, and DBpedia→AG News, with seeds 42–44. For each window and seed, two parent models learn the focal old task from the same archived history prefix: one uses GradResp (T), and the other uses a scale-matched Random reader (R). Each trained parent then branches into two children, giving TT, TR, RT, and RR. The first letter denotes the parent; the second denotes the latest old task's reader copy used for future historical protection. Only this external protection copy is exchanged. The learned forward readers and writers, and the earlier protection prefix, remain fixed within each parent pair. The child reader uses the same raw GradResp reference, computed from the frozen backbone and an identical training-only probe sample, and is projected against its own protection state; its Gram preconditioner is then rebuilt. Thus the intervention changes the inherited protection state while preserving the learned parent model's forward computation at the start of each branch.

Training and verification. We use rank 8, LoRA scaling parameter $\alpha = 32$ (forward scale $\alpha/r = 4$), dropout 0.1, one training epoch per task, learning rate 10^{-3}, effective batch size 64, Gram preconditioning, and per-update optimizer-state reset. Training uses BF16 autocast and evaluation uses FP32. Historical filtering retains the implemented full FP32 QR boundary, with stable projection and row-norm restoration. Within each parent pair, all-parameter hashes match before child-reader installation. After installation, learned historical parameter hashes and zero-update probe logits, generated labels, NLL, and heldout accuracy agree; the installed child readers can differ while their writers are zero. Zero-update precision checks confirm this agreement. Paired branches use the same data order. Throughout training, historical parameters and the current reader remain frozen, Gram matrices contain finite entries, and optimizer state is reset once per update. All nine prescribed window–seed units, comprising 18 parent and 36 child training runs, were completed and included in the analysis.

Outcomes and aggregation. We compare T protection with R protection within the same parent: $\Delta = \text{outcome}_{T\text{ protection}} - \text{outcome}_{R\text{ protection}}$. Negative NLL differences and positive accuracy differences favor T protection. For the focal old and new tasks, we evaluate their complete benchmark test splits. These contain 56/638 examples for CB/WiC, 277/7,600 for RTE/IMDB, and 7,600/7,600 for DBpedia/AG News. We also evaluate all previously learned tasks on fixed dev subsets (27 examples for CB and 64 for each other task). Gradient diagnostics are evaluation-only measurements on disjoint old-task samples. Because initial functions agree within each parent pair, endpoint differences also equal differences in damage from that common starting function. NLL is averaged over valid target tokens within each example and then over examples; NLL effects remain separate by task. Tables 45 and 46 report paired effects, with means and sample standard deviations computed across the three training seeds within each window and parent.

Table 45: Effects of inherited protection on complete test splits.

Parent	n	Δ old NLL	Δ new NLL	Δ old Acc.	Δ new Acc.
CB→WiC					
T	3	-0.1383 ± 0.2238	-0.0025 ± 0.0439	$+11.90 \pm 20.62$	$+0.84 \pm 9.98$
R	3	-0.0841 ± 0.1062	$+0.0078 \pm 0.0155$	$+5.95 \pm 5.74$	-1.41 ± 3.42
RTE→IMDB					
T	3	$+0.0023 \pm 0.0065$	$+0.0053 \pm 0.0111$	-0.60 ± 1.10	-0.61 ± 0.40
R	3	$+0.0007 \pm 0.006$	-0.0014 ± 0.0082	$+0.12 \pm 0.91$	-0.14 ± 0.64
DBpedia→AG News					
T	3	$+0.0001 \pm 0.0014$	-0.0005 ± 0.0219	$+0.07 \pm 0.05$	$+0.29 \pm 1.04$
R	3	$+0.0047 \pm 0.0054$	$+0.0049 \pm 0.0131$	-0.01 ± 0.10	$+0.29 \pm 0.27$

T: GradResp; R: scale-matched Random. Effects subtract R protection from T protection within the same parent. Accuracy differences are in percentage points; NLL differences are task-specific.

Window-specific results. CB→WiC provides the clearest retention effect: T protection lowers complete-test old-task NLL in all six parent–seed comparisons. Mean old-task accuracy changes are $+11.90$ and $+5.95$ pp for T and R parents, respectively. Their magnitudes vary strongly across seeds. For T parents, the three old-task accuracy differences are 0, 0, and $+35.71$ pp; for R parents they are $+3.57$, $+1.79$, and $+12.50$ pp. The largest retention gains occur at seed 44, where new-

Table 46: Complete-test protection effects for every window, parent, and seed.

Parent	Seed	Δ old NLL	Δ new NLL	Δ old Acc.	Δ new Acc.
CB→WiC					
T	42	-0.0044	+0.0382	+0.00	+9.87
R	42	-0.0207	+0.0254	+3.57	+0.94
T	43	-0.0139	+0.0033	+0.00	+2.51
R	43	-0.0248	-0.0034	+1.79	+0.16
T	44	-0.3966	-0.049	+35.71	-9.87
R	44	-0.2067	+0.0013	+12.50	-5.33
RTE→IMDB					
T	42	+0.0098	+0.0002	-1.81	-0.29
R	42	-0.0031	+0.0077	+0.00	-0.88
T	43	-0.0016	+0.018	+0.36	-1.05
R	43	-0.0024	-0.0083	+1.08	+0.13
T	44	-0.0014	-0.0025	-0.36	-0.47
R	44	+0.0077	-0.0038	-0.72	+0.32
DBpedia→AG News					
T	42	+0.0007	-0.0254	+0.05	+1.42
R	42	+0.0017	+0.0099	+0.07	+0.05
T	43	+0.0012	+0.0087	+0.03	+0.07
R	43	+0.0109	+0.0148	-0.13	+0.25
T	44	-0.0015	+0.0153	+0.13	-0.62
R	44	+0.0014	-0.0099	+0.03	+0.58

All differences use T protection minus R protection from the same parent function. Accuracy is in percentage points. These rows are paired effects, with seeds as training repeats; parent and window comparisons are paired within each seed.

task accuracy decreases by 9.87 and 5.33 pp. The small CB test split and this acquisition–retention trade-off contextualize the window means. For RTE→IMDB, mean complete-test old-task NLL increases by 0.0023 and 0.0007 under T protection, and accuracy effects have mixed signs. For DBpedia→AG News, focal old-task heldout NLL decreases in 6/6 comparisons, whereas complete-test old-task NLL increases in 5/6. Mean complete-test old-task accuracy changes are $+0.07$ and -0.01 pp for T and R parents. This window therefore shows sensitivity to both the sample used for evaluation and the metric used to assess old-task retention.

First-step interference and evaluation scope. The first-step diagnostics distinguish immediate interference from the endpoint of a full task. Here g is the unpreconditioned old-task gradient, and δW is the first actual effective weight increment, including the LoRA forward scale; their inner products sum over adapted layers. In all six RTE→IMDB parent–seed comparisons, T protection lowers both the old-task gradient–update inner product and the independently evaluated first-step old-task NLL increment. The same window has no mean complete-test old-task NLL benefit at the task endpoint. Table 47 reports both first-step quantities, focal old-task heldout NLL, and earlier-history accuracy. Earlier-history accuracy is averaged equally over tasks preceding the focal old task (one, six, or eleven tasks), then across seeds; these histories and the focal task share the same seed-level aggregation.

Table 47: Local interference and heldout historical outcomes.

Parent	$\Delta\langle g, \delta W\rangle$	Δ first-step NLL	Δ endpoint NLL	Earlier Acc.
CB→WiC				
T	$+0.011 \pm 0.397$	-0.234 ± 0.44	-163 ± 249.5	$+9.90 \pm 19.85$
R	-0.634 ± 0.62	$+0.164 \pm 0.99$	-109.6 ± 142.1	$+3.65 \pm 10.40$
RTE→IMDB				
T	-0.638 ± 0.233	-0.131 ± 0.051	$+1.953 \pm 4.448$	-0.05 ± 0.87
R	-0.384 ± 0.515	-0.728 ± 0.742	$+1.42 \pm 3.825$	$+0.05 \pm 0.63$
DBpedia→AG News				
T	-0.027 ± 1.644	$+0.035 \pm 0.062$	-0.024 ± 0.028	-0.04 ± 0.47
R	$+1.631 \pm 2.8$	-0.039 ± 0.034	-0.249 ± 0.11	-0.05 ± 0.46

The three middle columns are multiplied by 10^3. The gradient–update inner product and first-step loss increment concern the focal old task; endpoint NLL uses its heldout subset. Earlier Acc. is the equally weighted accuracy difference over earlier historical tasks, in percentage points. All contrasts use T protection minus R protection. Gradient and loss-evaluation samples are disjoint.

Interpretation. Changing inherited protection at a fixed learned function changes interference during the next task, with the clearest endpoint retention benefit on CB→WiC. The effect depends on the task window and parent model, and changes in new-task learning accompany some of the largest retention gains. This experiment identifies the protection state's effect through the subsequent reader, Gram geometry, and training trajectory. The update-deletion experiment in Appendix D.7 separately examines the contribution of unprotected old-task-sensitive update components.

D.9 Manipulating Reader Coverage

We select reader targets with lower or higher construction deficit using shared task-start training probes, then compare acquisition and subsequent damage. The RTE branches also continue through the remaining tasks to the end of the complete 15-task stream.

Construction and training. We fix three windows, CB→WiC, RTE→IMDB, and DBpedia→AG News, with seeds 42–44. Each starts from a shared checkpoint before the old task and branches into the reference reader and two coverage-targeted candidates. For an installed reader $A = U\Sigma V^\top$, we construct

$$A_\theta = U\Sigma(V\cos\theta + E\sin\theta)^\top, \qquad \theta = 30^\circ, \tag{61}$$

where E is orthonormal and orthogonal to both V and the historical space. This preserves rank and singular values. We generate 16 candidates per layer and select lower/higher construction-deficit targets using only the shared task-start training probe. The angle, candidate count, windows, and seeds are fixed independently of retention outcomes.

All writers start at zero. Each branch trains the old task, measures its actual task-end deficit, accuracy, and diagnostic NLL, then freezes and retains its own reader while training the next task. Branches follow the same algorithm and data order, each with its own historical projection and Gram matrix. This gives 27 two-task branches, or 54 task trainings. Branches share the initial model function, retain frozen historical factors, and use the same diagnostic backend and optimizer-state reset rule. Five predetermined optimizer updates per branch yield 135 local observations of effective weight changes, protection residuals, first-order harm, and independent loss changes.

Manipulation and outcomes. The primary task-end deficit is higher for the high-target candidate in all nine windows, by 1.40–5.65 pp (Table 48). Acquisition-accuracy differences are at most 1.79 pp. Acquisition NLL is less closely matched: the recorded 10% relative-difference diagnostic flag holds in five windows (CB seeds 42 and 44, and all RTE seeds), with CB/43 near 32% and all DBpedia windows above the threshold. The table reports absolute NLL differences for all windows, including the small DBpedia losses. Energy-weighted aggregation reverses the high/low deficit ordering for DBpedia seeds 43 and 44.

High-target branches have larger old-task accuracy drops after the next task in four windows, smaller drops in four, and a tie in one. For diagnostic NLL damage, the counts are three larger and six

Table 48: All coverage-intervention comparisons, high target minus low target.

			Acquisition		Next-task damage	
Old task	Seed	Deficit	Acc.	NLL	Acc. drop	NLL change
CB	42	+5.248	+1.79	−0.0041	+3.57	−0.00612
CB	43	+5.648	−1.79	+0.03536	+3.57	+0.01335
CB	44	+5.235	+1.79	−0.002382	−3.57	−0.1387
RTE	42	+4.335	+0.36	+0.00276	0.00	−0.004234
RTE	43	+3.872	+0.36	+0.003025	+0.72	−0.001225
RTE	44	+4.480	−1.08	−0.003097	+1.08	−0.005183
DBpedia	42	+3.334	−0.03	+0.005961	−0.11	−0.002601
DBpedia	43	+1.395	+0.12	+0.000065	−0.04	+0.001042
DBpedia	44	+4.510	+0.13	−0.004526	−0.04	+0.004214

Deficit and accuracy differences are in pp; NLL differences are unscaled. Damage is evaluated on the old task relative to each branch's own acquisition endpoint. A positive damage contrast means greater damage in the high-target branch. For RTE/42, the high-minus-low accuracy-drop differences round to zero at the numerical precision reported in the table.

smaller. CB/44 exhibits severe damage in all arms: reference accuracy changes from 94.64% to 44.64%, low target from 91.07% to 30.36%, and high target from 92.86% to 35.71%. Across the windows, separation of the primary deficit accompanies mixed directions of subsequent damage.

Continuation to the stream endpoint. The six low/high RTE branches continue from their own IMDB checkpoints through Yelp, Amazon, SST-2, DBpedia, AG News, MultiRC, and Yahoo, requiring 42 additional task trainings. Each branch trains these tasks using its own evolving history. Table 49 reports RTE retention through the 15-task stream. Seed 42 gives equal signed drops; high target is worse for seed 43 and improves more for seed 44. The retention ordering therefore varies across seeds despite the common direction of the task-end deficit contrast.

Table 49: RTE retention at the end of the 15-task stream after branch continuation.

Seed	Candidate	Acquisition	Stream end	Signed drop	Clipped forgetting
42	Low	87.00	85.56	+1.44	1.44
42	High	87.36	85.92	+1.44	1.44
43	Low	86.28	83.75	+2.53	2.53
43	High	86.64	82.67	+3.97	3.97
44	Low	85.20	85.56	−0.36	0.00
44	High	84.12	84.84	−0.72	0.00

Acquisition and stream-end accuracy are percentages on the full RTE test set. Signed drop is acquisition minus stream-end accuracy in pp; clipped forgetting is its positive part. Each branch trains all seven remaining tasks after IMDB.

D.10 Cross-Seed Response-State Diagnosis

To characterize the learned states behind the intervention outcomes in Tab. 1, we compare adapter-response subspaces across independent seeds. Let $V^{(a)}_{4,t,\ell}$ contain the leading four right singular vectors of the response $\Delta W X^\top$ for seed a, task t, and layer ℓ. For seeds $a \neq b$, define

$$o_{ab,t,\ell} := \frac{1}{4} \left\| (V^{(a)}_{4,t,\ell})^\top V^{(b)}_{4,t,\ell} \right\|_F^2 . \tag{62}$$

The reported response overlap averages $o_{ab,t,\ell}$ over the three seed pairs, four tasks, and 96 common self-attention query and value projection layers evaluated on fixed train-only probes. For the rank-8 adapters, the $k = 4$ statistic summarizes the leading half of the response space. The measurement covers self-attention, with the 48 adapted cross-attention matrices outside its scope. The common-94 comparison reports both $k = 4$ and $k = 8$.

Full FT serves as an independent geometric reference. The overlap increase shows that GradResp yields a more reproducible learned response organization on matched probes. This cross-seed

Table 50: Cross-seed response-state diagnosis across independent training runs on four tasks.

Method	Setting	LC ↑	Overlap $k = 4, 96$	Overlap $k = 4, 94$	Overlap $k = 8, 94$
O-LoRA	Random	78.5%	0.339	0.335	0.309
	GradResp	79.8%	0.699	0.701	0.855
GORP	Random	77.7%	0.356	0.353	0.316
	GradResp	79.3%	0.695	0.697	0.856
Full FT	Independent	–	0.910	–	–

LC reports means over three seeds (42–44). GradResp has higher mean cross-seed response overlap than Random for both O-LoRA and GORP. The first $k = 4$ column uses all 96 common projections; the two common-94 columns provide the $k \in \{4, 8\}$ sensitivity on a scope with sufficient numerical rank for every comparison. This fixed-probe diagnosis complements the reader-only outcome intervention reported in Tab. 1 for the 15-task sequence.

statistic measures subspace consistency; reader deficit measures projection error, and the controlled intervention measures retention outcomes.

D.11 Retained-State and Computational Costs

Persistent historical state consists of low-rank coefficients. For T5-large, excluding the frozen base model, the current rank-8 writer has 1,179,648 trainable coefficients, each retained reader–writer state has 2,359,296, and the accumulated adapter has 35,389,440 coefficients after 15 tasks. Stored historical examples, gradients, and features are all zero. For LLaMA 3.1–8B, the 32 layers have query output width 4,096 and value output width 1,024. At rank 8, each task adds 1,310,720 writer and 2,097,152 reader coefficients: 3,407,872 per block and 51,118,080 after 15 tasks. These factor counts exclude optimizer buffers and temporary probe workspaces. Per-layer setup, gradient-processing, and inference costs are given in Appendix B.6.