RG-LoRA: Residual Geometry-Guided Continual Low-Rank Adaptation

Anonymous authors
Paper under double-blind review

Abstract

Continual learning requires a model to acquire new tasks while retaining earlier knowledge, yet sequential adaptation can cause catastrophic forgetting. This challenge persists in low-rank adaptation, where task updates may interfere with previously learned representations. We ask whether past updates can themselves provide a bounded history for guiding future adaptation. We introduce *RG-LoRA*, a residual geometry-guided approach to continual low-rank adaptation. RG-LoRA summarizes each learned residual into a rank-bounded history that captures the directions and strengths of past model changes. This history attenuates new updates along strongly represented directions without preventing their reuse. After each task, the learned residual is merged into the adapted model, while its geometry updates the bounded history. Across continual classification and instruction-generation benchmarks, RG-LoRA is competitive with reported baselines on T5-Large and achieves substantially higher final performance on Llama-2-7B-Chat and Llama-3.2-3B, with larger gains on longer task sequences. These results suggest that residual geometry provides a compact and effective signal for continual low-rank adaptation without requiring a separate post-task data collector.

1 Introduction

Continual learning studies how a model can learn a sequence of tasks while retaining knowledge acquired from earlier tasks. Sequential training makes this difficult because updates for a new task can interfere with knowledge learned from previous tasks, leading to catastrophic forgetting. This problem can become more severe as updates accumulate over long task sequences. Updating the full model at every task is also computationally expensive, which makes parameter-efficient fine-tuning a practical choice for continual learning.

Low-rank adaptation (LoRA) Hu et al. (2022) represents each task update with a low-rank residual and provides an efficient way to adapt the model sequentially. However, low-rank updates can still interfere with changes introduced by previous tasks. We therefore study continual learning with LoRA and focus on how each new residual interacts with the updates accumulated from earlier tasks.

Existing continual LoRA methods handle this interaction in different ways. Subspace-based methods constrain successive task updates Wang et al. (2023); Liang & Li (2024); Lu et al. (2025); Wang et al. (2026), while regularization-based methods restrict changes along directions considered important to previously learned tasks Kirkpatrick et al. (2017); Das Biswas et al. (2026). Other methods retain task-specific adaptations or progressively merge them into a shared model Qiao & Mahdavi (2026). These methods differ mainly in how they carry information from previous tasks into later adaptation. Restricting previously used directions can limit their reuse when they remain useful for later tasks, while regularization-based methods require an additional mechanism to estimate which directions should be protected. Keeping task-specific adaptations preserves earlier updates explicitly, but the persistent state grows with the number of tasks. Continual merging avoids this growth by incorporating each update into a shared model. This leads to a key question: can residual updates learned on previous tasks serve as the historical signal needed to guide future LoRA adaptation?

We start from a simple observation: a learned LoRA residual is the result of task-specific optimization and records how that task changed the model. Although the residual does not recover previous data or objectives, its geometry captures the directions represented by the learned parameter change.

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.

This suggests that residuals learned on previous tasks can themselves provide a historical signal for guiding later updates.

We therefore propose RG-LoRA, a residual geometry-guided method for continual low-rank adaptation. RG-LoRA summarizes previous residual geometry in a bounded history and uses it to reduce new updates along strongly represented directions without preventing their reuse. After each task, the executed residual is merged into the backbone, and its geometry is used to update the history. The method requires neither replay, a separate post-task data collector, nor a growing adapter bank. We evaluate RG-LoRA on T5-Large and Llama models across continual classification and instruction-generation benchmarks. RG-LoRA remains competitive with reported baselines on T5-Large and achieves substantially higher final performance than reported baselines on the two Llama backbones.

2 Related Work

Subspace-based continual low-rank adaptation. A prominent direction in parameter-efficient continual learning controls the subspaces used by successive low-rank updates. O-LoRA encourages approximate orthogonality to previous update subspaces Wang et al. (2023), while InfLoRA identifies interference-free low-rank spaces in class-incremental vision Liang & Li (2024). CLoRA regularizes LoRA factors against predefined subspaces Lu et al. (2025), C-LoRA introduces routing with an orthogonality constraint Zhang et al. (2025), and SD-LoRA decouples magnitude and directional adaptation Wu et al. (2025). SplitLoRA separates historical gradient space into primary and minor subspaces Qiu et al. (2026), whereas ASO-LoRA applies similarity-weighted soft-orthogonality between task-specific LoRA subspaces Wang et al. (2026).

Structured direction and subspace adaptation. KeepLoRA projects new optimization into a residual subspace excluding principal parameter and previous-task feature directions Luo et al. (2026). LoDA separates general and task-specific subspaces using projection energy He et al. (2026), while E^2-LoRA uses output feature drift to concentrate knowledge into high-energy components and allocate rank dynamically Li et al. (2026). CoDyRA similarly adapts effective LoRA rank during continual learning Lu et al. (2024). RG-LoRA instead derives input- and output-side geometry directly from executed residuals, consolidates it into a bounded history, and uses this history for task-fixed soft attenuation without explicit subspace partitioning, routing, or dynamic rank allocation.

Bounded-state continual adaptation and merging. ELLA derives an elementwise regularizer from an aggregated historical update Das Biswas et al. (2026), while Merge Before Forget continually merges LoRA updates into a shared model Qiao & Mahdavi (2026). Task arithmetic provides an earlier approach to composing low-rank task updates Chitale et al. (2023), and Meta-UCF generates adaptations from task representations using a shared generator Xiao et al. (2026). RG-LoRA also maintains bounded persistent state, but separates the merged residual from a bounded residual-geometry history used only to guide future adaptation.

Importance-aware geometric protection. EWC uses Fisher information to protect important parameters Kirkpatrick et al. (2017), Riemannian Walk combines parameter importance with optimization-path information Chaudhry et al. (2018), and MAS estimates importance from function sensitivity Aljundi et al. (2018). SGP scales gradient components according to directional importance in historical subspaces derived from input representations Saha & Roy (2023). RG-LoRA instead constructs input- and output-side operators from the exact Gram matrices of executed low-rank residuals and uses them for task-fixed soft attenuation of subsequent LoRA residuals.

3 Problem Formulation and Design Rationale

3.1 Continual LoRA and Residual Interference

Consider a sequence of tasks $\mathcal{T}_1, \ldots, \mathcal{T}_T$ learned without revisiting data from previous tasks. For an adapted linear layer with weight $W_{t-1} \in \mathbb{R}^{d_o \times d_i}$ before task t, LoRA represents the task-specific update as a low-rank residual

$$D_t = sB_tA_t, \qquad A_t \in \mathbb{R}^{r \times d_i}, \quad B_t \in \mathbb{R}^{d_o \times r}, \tag{1}$$

where $r \ll \min(d_i, d_o)$ and s is the LoRA scaling factor. Here, D_t denotes the ordinary LoRA residual used to formulate the continual adaptation problem. RG-LoRA later distinguishes this residual from the executed residual $\widetilde{D}_t$, which includes the task-fixed attenuation maps. After learning task t, the adapted weight can be written as

$$W_t = W_{t-1} + D_t. \tag{2}$$

The objective is to learn each new task while preserving knowledge acquired from earlier tasks. Let $\mathcal{L}_j(W)$ denote the loss of a previous task $j < t$. The effect of a new residual on that task is

$$\Delta\mathcal{L}_j(D_t) = \mathcal{L}_j(W_{t-1} + D_t) - \mathcal{L}_j(W_{t-1}). \tag{3}$$

For previous tasks, however, the corresponding data are not retained. Consequently, the exact change in their losses is not directly available during subsequent adaptation. This motivates seeking a historical signal that can be derived from the updates themselves.

A useful historical signal should summarize how previous optimization changed the model while remaining available after the corresponding task data have been discarded. The learned residual naturally satisfies this requirement, which motivates examining what information about previous adaptation is contained in its geometry.

3.2 Optimization-Distilled Residual Geometry

To characterize what this geometry captures, we examine the functional disturbance induced by a learned residual. Consider an adapted linear projection

$$y = Wx \tag{4}$$

and a learned residual D. Applying the residual changes the layer output from y to $y + Dx$. For a loss ℓ with output gradient $g = \nabla_y \ell$, a first-order Taylor expansion gives

$$\ell(y + Dx) = \ell(y) + g^\top Dx + \mathcal{O}\big(\|Dx\|_2^2\big). \tag{5}$$

The corresponding first-order disturbance is therefore

$$\Delta\ell \approx g^\top Dx. \tag{6}$$

This expression reveals two complementary geometric views of the same disturbance. From the input side,

$$\big(g^\top Dx\big)^2 \le \|g\|_2^2 \|Dx\|_2^2 = \|g\|_2^2 x^\top D^\top Dx. \tag{7}$$

This gives the positive semidefinite operator

$$G_{\text{in}} = D^\top D, \tag{8}$$

whose quadratic form measures how strongly the residual acts along an input direction. Its dominant directions therefore identify parts of the input space that are most strongly represented by the learned update.

The same disturbance can be viewed from the output side:

$$\big(g^\top Dx\big)^2 \le \|x\|_2^2 \|D^\top g\|_2^2 = \|x\|_2^2 g^\top DD^\top g. \tag{9}$$

This gives the corresponding output-side operator

$$G_{\text{out}} = DD^\top, \tag{10}$$

whose dominant directions identify where the effect of the learned update is most strongly expressed in the output space. Together, G_{in} and G_{out} provide two complementary directional summaries of the same residual and form a natural historical signal for guiding subsequent adaptation.

To make this directional interpretation explicit, consider the singular value decomposition

$$D = U\Sigma V^\top. \tag{11}$$

Then

$$G_{\text{in}} = V\Sigma^2 V^\top, \qquad G_{\text{out}} = U\Sigma^2 U^\top. \tag{12}$$

This decomposition shows that G_{in} retains the right singular directions of the residual and G_{out} retains the left singular directions, while the squared singular values measure the strength associated with each direction. The decomposition is used here to interpret the geometry rather than to specify how that geometry is stored or updated.

Because D is the outcome of task optimization, G^{in} and G^{out} summarize the directional structure of the learned parameter change. We refer to this representation as optimization-distilled residual geometry and use it as the historical signal for guiding subsequent adaptation. In RG-LoRA, we compute these operators from the executed residual $\widetilde{D}_t$ defined in Section 4.

3.3 From Residual Geometry to Soft Attenuation

The residual geometry derived above identifies directions that were strongly represented by previous updates. Let H denote a generic historical operator formed from such residual geometry, and let u_i denote one of its directions. Its historical strength is measured by

$$h_i = u_i^\top H u_i. \tag{13}$$

A suitable attenuation rule should remain strictly positive and decrease with historical strength,

$$h_i > h_j \quad \Longrightarrow \quad \alpha(h_i) \le \alpha(h_j), \qquad \alpha(h) > 0. \tag{14}$$

Directions with stronger historical support are therefore attenuated more, while no direction is removed from the adaptation space.

We apply this attenuation on both sides of the residual. Let T and L denote positive-definite transformations induced by the input- and output-side historical operators, respectively. A guided residual can then be written as

$$\widetilde{D} = LDT. \tag{15}$$

The right transformation controls the input directions on which the residual acts, while the left transformation controls the output directions in which its effect is expressed.

RG-LoRA therefore uses residual geometry to guide future adaptation through soft two-sided attenuation rather than hard exclusion. Section 4 describes how the historical operators and attenuation maps are constructed.

4 Methodology

We propose RG-LoRA based on the hypothesis developed in Section 3 that the geometry of residual updates learned on previous tasks provides a useful historical signal for guiding subsequent LoRA adaptation. Rather than preventing future updates from reusing previously represented directions, RG-LoRA uses this signal to softly reduce changes along directions that have been strongly represented by earlier residuals. This controls interference while keeping previously used directions available for reuse.

For each adapted layer, RG-LoRA maintains separate input- and output-side residual-geometry histories. Before task t, these histories define fixed attenuation maps that shape a fresh LoRA residual during training. After the task is learned, the resulting executed residual is merged into the backbone and is also used to extract the residual geometry contributed by the current task. This geometry is consolidated into the bounded histories used for the next task. Figure 1 summarizes this continual adaptation cycle.

4.1 Bounded Residual-Geometry History

Consider an adapted layer $W \in \mathbb{R}^{d_o \times d_i}$. Before task t, RG-LoRA maintains an input-side historical operator $H_{t-1}^{\text{in}} \in \mathbb{R}^{d_i \times d_i}$ and an output-side historical operator $H_{t-1}^{\text{out}} \in \mathbb{R}^{d_o \times d_o}$. The input-side history summarizes feature directions on which previous residuals acted, whereas the output-side history summarizes directions in which their effects were expressed.

When the same construction applies to either side, we denote the corresponding history by $H_{t-1} \in \mathbb{R}^{d \times d}$. RG-LoRA stores it in the bounded form

$$H_{t-1} = U_{t-1} \operatorname{diag}\left(\lambda_{t-1} - \tau_{t-1}\mathbf{1}\right) U_{t-1}^\top + \tau_{t-1} I, \qquad \operatorname{rank}(U_{t-1}) \le R_H. \tag{16}$$

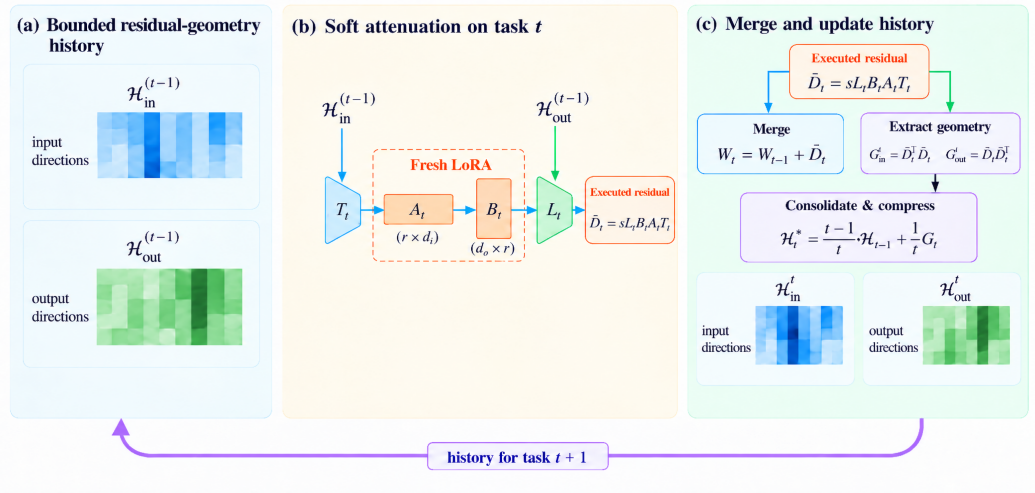


Figure 1: Overview of RG-LoRA. (a) Input- and output-side residual-geometry histories summarize directions represented by previous updates. (b) These histories induce task-fixed soft attenuation around a fresh LoRA update. (c) The executed residual is merged into the backbone, while its geometry is extracted and consolidated to form the histories for the next task. Heatmaps indicate relative directional strength.

The columns of U_{t-1} contain the residual directions retained explicitly, λ_{t-1} contains their spectral values, and $\tau_{t-1}I$ represents spectral mass whose orientation is not retained within the fixed budget. The dimension d equals d_i on the input side and d_o on the output side.

This representation keeps the historical state bounded independently of the number of observed tasks. It retains residual geometry rather than previous task adapters or previous task examples.

4.2 Task-Fixed Soft Attenuation

At the beginning of task t, define the mean spectral mass and normalized historical metric as

$$\bar{h} = \frac{\mathrm{Tr}(H_{t-1})}{d}, \qquad M(H_{t-1}) = I + \frac{H_{t-1}}{\bar{h}}. \tag{17}$$

For nonzero history, RG-LoRA uses the symmetric inverse square root

$$S(H_{t-1}) = M(H_{t-1})^{-1/2} = \left(I + \frac{H_{t-1}}{\bar{h}}\right)^{-1/2}. \tag{18}$$

This map satisfies $S(H_{t-1})^\top M(H_{t-1}) S(H_{t-1}) = I$, providing a symmetric normalization of the historical metric. An eigen-direction with value λ_i receives gain $(1 + \lambda_i/\bar{h})^{-1/2}$, so stronger historical directions are attenuated more while every gain remains strictly positive. When $\mathrm{Tr}(H_{t-1}) = 0$, we set $S(H_{t-1}) = I$.

The task-fixed maps are

$$T_t = S(H^{\mathrm{in}}_{t-1}), \qquad L_t = S(H^{\mathrm{out}}_{t-1}). \tag{19}$$

For a fresh raw rank-r LoRA residual $D_t = sB_tA_t$, the executed residual is

$$\widetilde{D}_t = sL_tB_tA_tT_t. \tag{20}$$

Here T_t acts on the input directions of the residual and L_t on its output directions.

To keep these maps from introducing additional factor-space preconditioning through backpropagation, the executed branch uses

$$\mathrm{IS}(X, \widetilde{X}) = X + \mathrm{stopgrad}(\widetilde{X} - X), \tag{21}$$

giving

$$\widetilde{D}_t = s\ \mathrm{IS}(B_t, L_t B_t)\ \mathrm{IS}(A_t, A_t T_t). \tag{22}$$

Its forward value is identical to Eq. (20), while the raw LoRA factors retain an identity Jacobian. Initialization details are given in Appendix C.4.

4.3 Merge and History Update

After training task t, the same executed residual $\widetilde{D}_t$ is used both to update the backbone and to construct the historical state for subsequent tasks. It is merged directly into the adapted backbone as

$$W_t = W_{t-1} + \widetilde{D}_t. \tag{23}$$

The same residual produces the input- and output-side geometry

$$G_t^{\mathrm{in}} = \widetilde{D}_t^\top \widetilde{D}_t, \qquad G_t^{\mathrm{out}} = \widetilde{D}_t \widetilde{D}_t^\top. \tag{24}$$

These are the two geometric views derived in Section 3.2. Since $\widetilde{D}_t$ has rank at most r, both operators admit compact spectral representations that can be obtained directly from the learned residual without revisiting task data.

The input- and output-side histories are updated independently. When the same construction applies to either side, let G_t denote the corresponding current-task operator. The new geometry is combined with the previous history through

$$H_t^\star = \frac{t-1}{t} H_{t-1} + \frac{1}{t} G_t. \tag{25}$$

Without compression, this recursion gives equal weight to the residual geometry contributed by each observed task.

To keep the persistent state bounded, RG-LoRA compresses $H_t^\star$ back to the representation in Eq. (16). The compression jointly considers the directions retained from the previous history and those introduced by the current residual, retains at most R_H explicit directions, and transfers spectral mass whose orientation is not retained into the isotropic component. The resulting state is written as

$$H_t = U_t\, \mathrm{diag}\,(\lambda_t - \tau_t \mathbf{1})\, U_t^\top + \tau_t I, \qquad \mathrm{rank}(U_t) \le R_H. \tag{26}$$

The compression preserves the trace of the consolidation target,

$$\mathrm{Tr}(H_t) = \mathrm{Tr}(H_t^\star). \tag{27}$$

The compact factorization, activity-aware mode ranking, and isotropic trace-closure procedure used to obtain U_t, λ_t, and τ_t are detailed in the appendix.

Applying the same update independently to G_t^{in} and G_t^{out} produces H_t^{in} and H_t^{out}. These histories define T_{t+1} and L_{t+1} before learning the next task. Once the backbone merge and history update are complete, the task-specific LoRA factors and temporary residual geometry are discarded. Only the continually updated backbone and the two bounded residual-geometry histories persist across tasks.

5 Experiments

We evaluate RG-LoRA across multiple model architectures and continual learning benchmarks to assess its effectiveness under different adaptation settings.

5.1 Experimental Setup

Benchmarks. We follow the six continual learning task orders used by O-LoRA Wang et al. (2023). O1–O3 correspond to the Standard CL (SC) benchmark, with four tasks per order, while O4–O6 are fifteen-task Long Sequence (LS) orders. We additionally evaluate on the fifteen-task subset of Super-NaturalInstructions (SuperNI) and its two task orders used by Merge Before Forget Qiao & Mahdavi (2026). These tasks cover classification, extraction, question answering, dialogue, and summarization Wang et al. (2022). Appendix D provides the full dataset composition, output spaces, evaluation metrics, and task orders.

Table 1: Final continual performance (%). RG-LoRA reports three-seed means, while baselines use published results. SC and LS average O1–O3 and O4–O6, respectively. SNI averages the two SuperNI orders. Bold and underline denote the best and second-best results within each backbone and column.

Backbone	Method	O1	O2	O3	SC	O4	O5	O6	LS	SNI-1	SNI-2	SNI
T5-Large	LwF (2017)	52.7	52.9	48.4	51.3	49.7	42.8	46.9	46.5	–	–	–
	EWC (2017)	46.3	45.3	52.1	47.9	44.9	44.0	45.4	44.8	–	–	–
	SeqLoRA (2023)	39.5	31.9	46.6	39.3	4.9	3.5	4.2	4.2	–	–	–
	O-LoRA (2023)	73.5	71.4	70.0	71.6	65.4	65.2	65.2	65.3	–	–	–
	LB-CL (2024)	76.9	76.5	76.8	76.7	68.4	67.3	71.8	69.2	–	–	–
	O-LoRA+MIGU (2024)	77.1	77.0	75.6	76.6	67.3	68.5	74.0	70.0	–	–	–
	ELLA (2026)	**80.0**	**80.0**	**79.8**	**79.9**	73.4	72.0	75.4	**73.6**	–	–	–
	ASO-LoRA (2026)	76.2	77.5	77.3	77.0	**73.7**	67.2	**77.4**	72.8	–	–	–
	RG-LoRA	79.91	79.93	79.27	79.70	70.94	**73.73**	74.80	73.16	–	–	–
Llama-2-7B-Chat	SeqLoRA (2023)	73.3	76.2	78.4	76.0	69.1	66.0	71.1	68.7	18.4	26.8	22.6
	O-LoRA (2023)	76.1	76.3	79.2	77.2	74.0	72.0	74.6	73.5	23.3	28.4	25.9
	CorDA (2024)	78.4	79.3	80.0	79.2	73.4	72.7	74.0	73.4	20.9	16.0	18.5
	MagMax (2024)	80.1	80.6	80.3	80.3	72.3	73.5	74.5	73.4	15.3	7.0	11.2
	KnOTS (2025)	67.9	65.9	70.8	68.2	61.5	60.1	58.0	59.9	34.6	30.1	32.4
	LoRA-LEGO (2025)	68.3	66.0	70.9	68.4	58.8	58.7	53.2	56.9	32.8	26.7	29.8
	MBF (2026)	80.1	80.8	80.4	80.4	75.0	74.4	75.1	74.8	38.7	35.7	37.2
	ASO-LoRA (2026)	75.0	74.8	78.1	76.0	59.1	64.5	59.2	60.9	–	–	–
	RG-LoRA	**81.12**	**81.13**	**80.83**	**81.03**	**81.36**	**81.60**	**81.64**	**81.53**	**52.29**	**51.64**	**51.96**
Llama-3.2-3B	MBF (2026)	74.3	75.8	75.3	75.1	73.3	72.5	74.9	73.6	32.7	34.6	33.7
	RG-LoRA	**81.50**	**81.52**	**80.87**	**81.29**	**79.80**	**80.33**	**80.59**	**80.24**	**50.20**	**52.72**	**51.46**

Models and baselines. We evaluate RG-LoRA on T5-Large Raffel et al. (2020), Llama-2-7B-Chat Touvron et al. (2023), and Llama-3.2-3B Grattafiori et al. (2024); Meta AI (2024). Unless otherwise specified, RG-LoRA uses LoRA rank 8, LoRA alpha 16, and history budget $R_H = 64$ explicit directions per layer side.

We train each CL task for one epoch and each SuperNI task for five epochs. For Llama models, O1–O3 use a learning rate of 2×10^{-5}, while O4–O6 use 10^{-4} because the long-sequence setting contains fewer optimization steps per task. These learning rates are fixed across the corresponding task orders. Appendix C provides the full hyperparameters and implementation details. All RG-LoRA runs and ablations use the gradient-aligned zero-function LoRA initialization described in Appendix C.4.

Published baseline results are taken from ELLA Das Biswas et al. (2026) for T5-Large, Merge Before Forget Qiao & Mahdavi (2026) for Llama models, and ASO-LoRA Wang et al. (2026). We use the published baseline results reported for the corresponding task orders and compare RG-LoRA against the strongest reported performance available in those results. RG-LoRA results are reported over three training seeds (41, 42, and 43). All experiments were conducted on two NVIDIA RTX 4090 GPUs.

Metrics. For a stream with T tasks, let $a_{T,j}$ denote the score on task j after the complete task sequence. We report final average performance as

$$\mathrm{ACC} = \frac{1}{T} \sum_{j=1}^{T} a_{T,j}. \tag{28}$$

Classification tasks use normalized exact-match accuracy. For SuperNI, classification tasks use accuracy and generation tasks use ROUGE-L, with scores averaged equally across tasks. We additionally report backward transfer (BWT) and average forgetting on selected streams to evaluate retention, with their definitions provided in Appendix C. Full sampling and evaluation details are also provided in the appendix.

5.2 Main Results

Table 1 reports final continual performance, comparing three-seed RG-LoRA means with published baselines. Table 2 reports the corresponding variation across seeds. To complement final performance, Table 3 reports backward transfer and average forgetting on selected task streams.

Table 2: Three-seed RG-LoRA performance by task order (%). Results are reported as mean ± standard deviation.

Backbone	Stream	Order 1	Order 2	Order 3	Avg.
T5-Large	SC (O1–O3)	**79.91** ± 0.33	**79.93** ± 0.20	**79.27** ± 1.22	**79.70** ± 0.53
	LS (O4–O6)	**70.94** ± 0.65	**73.73** ± 2.51	**74.80** ± 1.07	**73.16** ± 0.73
Llama-2-7B-Chat	SC (O1–O3)	**81.12** ± 0.39	**81.13** ± 0.10	**80.83** ± 0.42	**81.03** ± 0.26
	LS (O4–O6)	**81.36** ± 1.19	**81.60** ± 0.39	**81.64** ± 0.52	**81.53** ± 0.53
	SNI (O1–O2)	**52.29** ± 0.42	**51.64** ± 0.67	–	**51.96** ± 0.52
Llama-3.2-3B	SC (O1–O3)	**81.50** ± 0.11	**81.52** ± 0.26	**80.87** ± 0.47	**81.29** ± 0.22
	LS (O4–O6)	**79.80** ± 0.10	**80.33** ± 0.64	**80.59** ± 0.22	**80.24** ± 0.27
	SNI (O1–O2)	**50.20** ± 0.58	**52.72** ± 0.99	–	**51.46** ± 0.48

Table 3: Backward transfer (BWT, higher is better) and average forgetting (Fgt, lower is better) on selected seed-42 task streams.

	CL O1		CL O4		SuperNI O1	
Backbone	BWT ↑	Fgt ↓	BWT ↑	Fgt ↓	BWT ↑	Fgt ↓
T5-Large	−0.86	1.04	−6.81	8.25	–	–
Llama-2-7B-Chat	+0.23	0.15	+0.15	1.35	−2.68	3.53
Llama-3.2-3B	−0.18	0.19	−0.14	1.29	−5.15	6.13

T5-Large. RG-LoRA achieves 79.70 ± 0.53 on SC and 73.16 ± 0.73 on LS, remaining competitive with the published ELLA results of 79.9 and 73.6, respectively. On the selected O4 trajectory, RG-LoRA shows 8.25 points of average forgetting.

Llama-2-7B-Chat. RG-LoRA achieves 81.03 ± 0.26 on SC, 81.53 ± 0.53 on LS, and 51.96 ± 0.52 on SuperNI. These results exceed the published Merge Before Forget baseline by 0.63, 6.73, and 14.76 points, respectively. The reported performance gap is larger on LS and SuperNI than on SC. On the selected CL trajectories, BWT remains slightly positive, whereas SuperNI exhibits more noticeable forgetting.

Llama-3.2-3B. RG-LoRA reaches 81.29 ± 0.22 on SC, 80.24 ± 0.27 on LS, and 51.46 ± 0.48 on SuperNI. Compared with Merge Before Forget, the gains are 6.19, 6.64, and 17.76 points, respectively. The reported performance gap also remains substantial on both LS and SuperNI for the smaller Llama-3.2-3B backbone. Performance is also stable across the three SC and LS task orders, with only small variation in the three-seed means.

5.3 Analysis

Behavior across model architectures. The largest reported performance gaps relative to published baselines occur on the two Llama backbones, particularly on LS and SuperNI. T5-Large shows weaker long-sequence retention. One possible reason is that its adapted projections span encoder self-attention, decoder self-attention, and cross-attention, whereas RG-LoRA summarizes residual geometry locally at each projection. Controlled experiments are needed to test this explanation.

Input- and output-side history. Table 4 shows that input-only attenuation nearly matches full RG-LoRA, whereas output-only attenuation is substantially weaker, particularly on SuperNI. This suggests that the input-side geometry carries the more stable historical signal. It records the feature directions on which previous residuals acted, while output-side geometry is more closely tied to how those changes were expressed on the output side of each adapted projection. The latter can vary more across heterogeneous tasks and output spaces. Output history therefore acts as a task-dependent complement rather than an equally strong independent source of protection.

Table 4: Effect of input- and output-side attenuation on Llama-2-7B-Chat for seed 42 (%). Input-only and output-only retain one attenuation side, while no-history disables both.

Variant	CL O1	CL O4	SuperNI O1
Full RG-LoRA	81.33	**80.13**	**52.03**
Input-only	**81.44**	80.12	51.59
Output-only	79.42	75.38	37.41
No-history	78.79	72.90	38.00

History and adapter capacity. The sensitivity analyses in Appendices E.4 and E.5 show that performance varies only modestly with the history budget on CL O1 and O4, whereas SuperNI benefits more from additional historical capacity (Table 15). Increasing R_H from 64 to 128 also improves T5 LS by 0.74 points (Table 14), suggesting that longer or more heterogeneous task streams may benefit from retaining more residual directions. By contrast, increasing the LoRA rank does not yield consistent improvements across the three settings (Table 16), indicating that greater adapter capacity alone does not resolve continual interference.

6 Conclusion

Continual LoRA requires learning new tasks while limiting interference with knowledge accumulated from earlier tasks. We proposed RG-LoRA, which uses the geometry of previously learned LoRA residuals as a bounded historical signal for continual adaptation. The method softly attenuates future updates along strongly represented directions while keeping those directions available for reuse. On T5-Large, RG-LoRA remains competitive with published baselines. On Llama-2-7B-Chat and Llama-3.2-3B, it achieves substantially higher final performance than the published baselines used in our evaluation.

RG-LoRA retains only a compressed summary of learned updates. Residual geometry does not recover old-task losses or functional sensitivity, and its Gram representation discards sign, input–output pairing, and contextual interactions. Bounded compression may also remove useful directions as task streams grow, while the current BWT and forgetting measurements cover selected single-seed trajectories.

A remaining challenge is how to preserve richer historical information without increasing persistent state with the number of tasks. We plan to study richer bounded histories that capture cross-module information and support selective updates when previous tasks are revisited with new data. We will also study adaptive consolidation for improving long-term retention while preserving the bounded-state property of RG-LoRA.

AI use statement

We used generative AI tools to aid writing and polishing, literature retrieval and discovery, and research ideation and execution during manuscript preparation. We reviewed and verified all AI-assisted content, including the methodology, mathematical derivations, experiments, citations, and reported results, and take full responsibility for the final manuscript.

Ethics statement

This work studies continual adaptation methods using established language benchmarks and introduces no new human-subject data. As with other model adaptation techniques, the resulting models may inherit biases, unsafe behaviors, or other limitations from the pretrained models and sequential training data. Improvements on the evaluated benchmarks should therefore not be interpreted as evidence of deployment safety or broader reliability.

Reproducibility statement

Appendix C specifies the model, optimizer, sampling, prompt, and evaluation settings. Appendix D lists the task orders and evaluation metrics, while Appendix F provides the mathematical and numerical checks used in our implementation. Code will be publicly released upon acceptance to support reproducibility.

References

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 139–154. Springer, 2018. doi: 10.1007/978-3-030-01219-9_9. URL https://www.ecva.net/papers/eccv_2018/papers_ECCV/html/Rahaf_Aljundi_Memory_Aware_Synapses_ECCV_2018_paper.php.

Arslan Chaudhry, Puneet K. Dokania, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 532–547. Springer, 2018. doi: 10.1007/978-3-030-01252-6_33. URL https://openaccess.thecvf.com/content_ECCV_2018/html/Arslan_Chaudhry__Riemannian_Walk_ECCV_2018_paper.html.

Rajas Chitale, Ankit Vaidya, Aditya Manish Kane, and Archana Ghotkar. Task arithmetic with LoRA for continual learning. In *Workshop on Advancing Neural Network Training: Computational Efficiency, Scalability, and Resource Optimization (WANT@NeurIPS 2023)*, 2023. URL https://openreview.net/forum?id=4CLNFKi12w.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300/.

Shristi Das Biswas, Yue Zhang, Anwesan Pal, Radhika Bhargava, and Kaushik Roy. ELLA: Efficient lifelong learning for adapters in large language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1907–1924. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.eacl-long.84. URL https://aclanthology.org/2026.eacl-long.84/.

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 2024*, pp. 6503–6522. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.379. URL https://aclanthology.org/2024.findings-emnlp.379/.

Aaron Grattafiori et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Lingfeng He, De Cheng, Huaijie Wang, Xi Yang, Nannan Wang, and Xinbo Gao. Task-driven subspace decomposition for knowledge sharing and isolation in LoRA-based continual learning. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=UJJfkcJ7K9.

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

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences.

In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 252–262. Association for Computational Linguistics, 2018. doi: 10.18653/v1/N18-1023. URL https://aclanthology.org/N18-1023/.

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

Longhua Li, Lei Qi, Qi Tian, and Xin Geng. Energy-structured low-rank adaptation for continual learning. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=4OP391X7Qm.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017. doi: 10.1109/TPAMI.2017.2773081. URL https://doi.org/10.1109/TPAMI.2017.2773081.

Yan-Shuo Liang and Wu-Jun Li. Inflora: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23638–23647, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Liang_InfLoRA_Interference-Free_Low-Rank_Adaptation_for_Continual_Learning_CVPR_2024_paper.html.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pp. 74–81. Association for Computational Linguistics, 2004. URL https://aclanthology.org/W04-1013/.

Haodong Lu, Chongyang Zhao, Minhui Xue, Lina Yao, Kristen Moore, and Dong Gong. Adaptive rank, reduced forgetting: Knowledge retention in continual learning vision-language models with dynamic rank-selective LoRA. *arXiv preprint arXiv:2412.01004*, 2024. URL https://arxiv.org/abs/2412.01004.

Yuheng Lu, Bingshuo Qian, Caixia Yuan, Huixing Jiang, and Xiaojie Wang. Controlled low-rank adaptation with subspace regularization for continued training on large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19165–19181. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.940. URL https://aclanthology.org/2025.acl-long.940/.

Mao-Lin Luo, Zi-Hao Zhou, Yi-Lin Zhang, Yuanyu Wan, Min-Ling Zhang, and Tong Wei. KeepLoRA: Continual learning with residual gradient adaptation. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/dc3ca8bcd613e43ce540352b58d55d6d-Abstract-Conference.html.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150. Association for Computational Linguistics, 2011. URL https://aclanthology.org/P11-1015/.

Daniel Marczak, Bartlomiej Twardowski, Tomasz Trzcinski, and Sebastian Cygert. MagMax: Leveraging model merging for seamless continual learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 379–395, 2024. doi: 10.1007/978-3-031-73013-9_22. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/papers/11489.pdf.

Meta AI. Llama 3.2 model card. Model card, 2024. URL https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md.

Mohammad Taher Pilehvar and Jose Camacho-Collados. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1267–1273. Association for Computational Linguistics, 2019. doi: 10.18653/v1/N19-1128. URL https://aclanthology.org/N19-1128/.

Fuli Qiao and Mehrdad Mahdavi. Learn more, but bother less: Parameter efficient continual learning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 97476–97498, 2024. doi: 10.52202/079017-3092. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/b0bc711f48724237b38823c4d9cee10b-Abstract-Conference.html.

Fuli Qiao and Mehrdad Mahdavi. Merge before forget: A single lora continual learning via continual merging. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0dedca10992d34035c9a9776254cfa8c-Abstract-Conference.html.

Haomiao Qiu, Miao Zhang, Ziyue Qiao, Weili Guan, Min Zhang, and Liqiang Nie. Splitlora: Balancing stability and plasticity in continual learning through gradient space splitting. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/5035a409f5798e188079e236f437e522-Abstract-Conference.html.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Gobinda Saha and Kaushik Roy. Continual learning with scaled gradient projection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8):9677–9685, 2023. doi: 10.1609/aaai.v37i8.26157.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642. Association for Computational Linguistics, 2013. URL https://aclanthology.org/D13-1170/.

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with SVD to tie the knots. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/0d4f8a5109c5083b5307fcd0bddae7a7-Abstract-Conference.html.

Hugo Touvron et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. URL https://arxiv.org/abs/2307.09288.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. SuperGLUE: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems*, volume 32, pp. 3261–3275, 2019a.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=rJ4km2R5t7.

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 2023*, pp. 10658–10671. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.715. URL https://aclanthology.org/2023.findings-emnlp.715/.

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 *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18472–18487. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.842. URL https://aclanthology.org/2026.acl-long.842/.

Yizhong Wang et al. Super-naturalinstructions: Generalization via declarative instructions on 1600+ nlp tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5085–5109. Association for Computational Linguistics, 2022. doi: 10.18653/v1/2022.emnlp-main.340. URL https://aclanthology.org/2022.emnlp-main.340/.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122. Association for Computational Linguistics, 2018. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101/.

Yichen Wu, Hongming Piao, Long-Kai Huang, Renzhen Wang, Wanhua Li, Hanspeter Pfister, Deyu Meng, Kede Ma, and Ying Wei. SD-LoRA: Scalable decoupled low-rank adaptation for class incremental learning. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/92f43b1d33fae4aa1958f75317f0cec1-Abstract-Conference.html.

ShiLin Xiao, Tianxiang Xu, Canran Xiao, Weihao Luo, Liwei Hou, and Chuangxin Zhao. Meta-UCF: Unified task-conditioned lora generation for continual learning in large language models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/12202970782399ee67981dc5269c3b8a-Abstract-Conference.html.

Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen Leon Song, Jianlong Wu, Liqiang Nie, and Bernard Ghanem. CorDA: Context-oriented decomposition adaptation of large language models for task-aware parameter-efficient fine-tuning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 71768–71791, 2024. doi: 10.52202/079017-2292. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/83f95bb0ac5046338ea2afe3390e9f4b-Abstract-Conference.html.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28, pp. 649–657, 2015. URL https://papers.nips.cc/paper/2015/hash/250cf8b51c773f3f8dc8b4be867a9a02-Abstract.html.

Xin Zhang, Liang Bai, Xian Yang, and Jiye Liang. C-LoRA: Continual low-rank adaptation for pre-trained models. *arXiv preprint arXiv:2502.17920*, 2025. URL https://arxiv.org/abs/2502.17920.

Ziyu Zhao, Tao Shen, Didi Zhu, Zexi Li, Jing Su, Xuwu Wang, and Fei Wu. Merging LoRAs like playing LEGO: Pushing the modularity of LoRA to extremes through rank-wise clustering. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/b54e0146a82945f01e69c2e3309ba925-Abstract-Conference.html.

A Detailed Derivations

This appendix gives the algebra and implementation details used by RG-LoRA. All statements apply independently to each adapted linear layer, and the layer index is omitted when it is not needed.

A.1 Connection to First-Order Functional Disturbance

Section 3.2 relates residual geometry to the local functional disturbance induced by a learned update. The derivation is given in more detail here. Consider a linear layer $y = Wx$ and an executed residual $D \in \mathbb{R}^{d_o \times d_i}$. For a twice-differentiable per-example loss $\ell(y)$, let $g = \nabla_y \ell$ and $H_y = \nabla_y^2 \ell$. Taylor expansion around the pre-update output gives

$$\Delta\ell = \ell(y + Dx) - \ell(y) = g^\top Dx + \frac{1}{2}(Dx)^\top H_y (Dx) + o(\|Dx\|_2^2). \tag{29}$$

The first-order term is therefore

$$\Delta\ell \approx g^\top Dx. \tag{30}$$

Applying Cauchy–Schwarz from the two sides yields

$$\left(g^\top Dx\right)^2 \le \|g\|_2^2 \, x^\top D^\top Dx, \tag{31}$$

$$\left(g^\top Dx\right)^2 \le \|x\|_2^2 \, g^\top DD^\top g. \tag{32}$$

Thus $D^\top D$ and $DD^\top$ describe the input- and output-side directions through which the learned residual can induce local change. If $D = U\Sigma V^\top$, then

$$D^\top D = V\Sigma^2 V^\top, \qquad DD^\top = U\Sigma^2 U^\top, \tag{33}$$

so the two operators retain the right and left singular directions of the same residual together with their squared strengths. RG-LoRA uses these two operators as an optimization-distilled historical signal. They summarize the learned update rather than the exact old-task loss landscape.

A.2 Properties of Residual Gram Geometry

Let $D \in \mathbb{R}^{d_o \times d_i}$ be the final executed task update from Eq. (20), not the raw factor product, and define $G^{\text{in}} = D^\top D$ and $G^{\text{out}} = DD^\top$.

Proposition 1 (PSD, rank, and trace)**.** *The residual Gram operators obey*

$$G^{\text{in}} \succeq 0, \qquad G^{\text{out}} \succeq 0, \tag{34}$$

$$\operatorname{rank}(G^{\text{in}}) = \operatorname{rank}(G^{\text{out}}) = \operatorname{rank}(D), \tag{35}$$

$$\operatorname{Tr}(G^{\text{in}}) = \operatorname{Tr}(G^{\text{out}}) = \|D\|_F^2. \tag{36}$$

Proof. For arbitrary x and z, $x^\top D^\top Dx = \|Dx\|_2^2 \ge 0$ and $z^\top DD^\top z = \|D^\top z\|_2^2 \ge 0$, proving positive semidefiniteness. If $D = U\Sigma V^\top$ is a thin SVD, then $D^\top D = V\Sigma^2 V^\top$ and $DD^\top = U\Sigma^2 U^\top$. The two operators therefore have the same nonzero eigenvalues and rank. Finally, cyclicity of the trace gives $\operatorname{Tr}(D^\top D) = \operatorname{Tr}(DD^\top)$, and either side equals the sum of squared entries of D. □

Proposition 2 (Complementarity of the two Gram operators)**.** *Although D contains both sides of the update, in general neither $D^\top D$ nor $DD^\top$ determines the directional basis of the other.*

Proof. For any orthogonal $O_o \in \mathbb{R}^{d_o \times d_o}$, define $D' = O_o D$. Then

$$D'^\top D' = D^\top D, \qquad D'D'^\top = O_o DD^\top O_o^\top, \tag{37}$$

which generally rotates the output basis while leaving the input Gram exactly unchanged. Conversely, right multiplication by an orthogonal input rotation can preserve the output Gram while rotating the input Gram. □

The two Gram operators summarize different sides of the same residual. $D^\top D$ gives the input-side directions, while $DD^\top$ gives the output-side directions. They share the same nonzero spectrum, but their eigenvectors belong to different spaces. Keeping only one therefore gives a one-sided variant of the two-sided guidance in Section 3.3.

Since $D = sL(BA)T$ and the LoRA factors have inner dimension r, $\operatorname{rank}(D) \le r$. Thus each task introduces at most r candidate directions on either layer side before historical compression.

Proposition 3 (LoRA factorization invariance)**.** *For any invertible $Z \in \mathbb{R}^{r\times r}$, replacing B by BZ and A by $Z^{-1}A$ leaves the RG-LoRA residual geometry unchanged.*

Proof. The product $(BZ)(Z^{-1}A) = BA$ is unchanged, so the executed residual and both Gram operators are unchanged. RG-LoRA therefore does not depend on a particular low-rank factorization of the same update. □

A.3 Compact SVD from the LoRA Factors

Dense formation of D, $D^\top D$, or $DD^\top$ is not needed. Define the attenuated factors $\widehat{B} = sLB$ and $\widehat{A} = AT$, so that $D = \widehat{B}\widehat{A}$ without redefining the raw LoRA factors. Thin QR factorizations give

$$\widehat{B} = Q_o R_o, \qquad \widehat{A}^\top = Q_i R_i, \tag{38}$$

and hence

$$D = Q_o(R_o R_i^\top)Q_i^\top. \tag{39}$$

Let the SVD of the $r \times r$ core be $R_o R_i^\top = P\Sigma V_c^\top$. Since Q_o, Q_i, P, V_c have orthonormal columns,

$$D = (Q_o P)\Sigma(Q_i V_c)^\top \tag{40}$$

is a thin SVD. Consequently, $(Q_i V_c, \Sigma^2)$ is the exact compact eigensystem of $D^\top D$ and $(Q_o P, \Sigma^2)$ is that of $DD^\top$. The decomposition therefore uses only the LoRA-sized core rather than a full layer matrix.

A.4 Task-Balanced Online Consolidation

Let G_j denote either the input- or output-side residual geometry contributed by task j. Starting from $H_1 = G_1$, the uncompressed recursion

$$H_t = \frac{t-1}{t}H_{t-1} + \frac{1}{t}G_t \tag{41}$$

produces an equal-task mean.

Proposition 4 (Equal-task history)**.** *If no compression is applied, then*

$$H_t = \frac{1}{t}\sum_{j=1}^{t} G_j. \tag{42}$$

Proof. The statement holds directly for $t = 1$. Assuming it holds at task $t-1$,

$$\frac{t-1}{t}H_{t-1} + \frac{1}{t}G_t = \frac{t-1}{t}\frac{1}{t-1}\sum_{j=1}^{t-1} G_j + \frac{1}{t}G_t = \frac{1}{t}\sum_{j=1}^{t} G_j. \tag{43}$$

□

Section 4.3 uses the same recursion to define the consolidation target $H_t^\star$. The bounded implementation described below compresses this target back to at most R_H explicitly represented directions.

Write the previous history as

$$H_{t-1} = U_o \operatorname{diag}(\lambda_o - \tau_o \mathbf{1})\, U_o^\top + \tau_o I, \tag{44}$$

and write the current residual geometry in compact spectral form as

$$G_t = U_g \operatorname{diag}(\lambda_g) U_g^\top. \tag{45}$$

The current-task operator G_t is the exact residual Gram matrix, so it has no incoming isotropic floor. With

$$\alpha_t = \frac{t-1}{t}, \qquad \beta_t = \frac{1}{t}, \tag{46}$$

the directional part of the consolidation target can be represented by

$$F_t = \left[\sqrt{\alpha_t}\, U_o \operatorname{diag}\left(\lambda_o - \tau_o \mathbf{1}\right)^{1/2},\ \sqrt{\beta_t}\, U_g \operatorname{diag}\left(\lambda_g\right)^{1/2} \right], \tag{47}$$

while the shared isotropic component is

$$\tau_b = \alpha_t \tau_o. \tag{48}$$

Let

$$F_t = \widehat{U}\widehat{\Sigma}V^\top \tag{49}$$

be its compact singular-value decomposition. If $\widehat{\sigma}_i$ is the i-th singular value and $\widehat{u}_i$ is the corresponding column of $\widehat{U}$, then the consolidation target has candidate spectral pairs

$$\left(\widehat{u}_i, \widehat{\lambda}_i\right), \qquad \widehat{\lambda}_i = \tau_b + \widehat{\sigma}_i^2. \tag{50}$$

The previous and current directions are therefore represented in one common eigensystem rather than in separate task-specific slots.

A.5 Residual Activity and Trace Closure

The bounded history can retain at most R_H candidate directions from Eq. (50). Current residual activity is used only to rank candidates with similar spectral importance.

For an input-side candidate $\widehat{u}_i$, its activity is

$$e_i^{\text{in}} = \left\| \widetilde{D}_t \widehat{u}_i \right\|_2^2. \tag{51}$$

For an output-side candidate $\widehat{v}_i$, the corresponding activity is

$$e_i^{\text{out}} = \left\| \widehat{v}_i^\top \widetilde{D}_t \right\|_2^2. \tag{52}$$

The two sides are ranked independently, so equal indices do not imply an input–output pairing.

For either side, let e_i denote the corresponding activity and let $\bar{e}$ denote the mean positive activity. The bounded activity multiplier is

$$a_i = 1 + \frac{e_i}{e_i + \bar{e}} \in [1, 2). \tag{53}$$

If all candidate activities are zero, a_i is set to 1 for every candidate.

Each candidate receives the priority

$$s_i = \widehat{\lambda}_i a_i^{0.25}. \tag{54}$$

The exponent keeps the activity term bounded, so spectral strength remains the main ranking signal. The retained index set contains the highest-priority candidates,

$$\mathcal{K} = \operatorname{TopK}_{R_H}\left(\{s_i\}\right), \tag{55}$$

before the isotropic consistency check described below. The priority score is used only to select directions and does not replace the spectral values $\widehat{\lambda}_i$.

After selecting the retained set, spectral mass whose orientation is not represented explicitly is assigned to the isotropic component

$$\tau_t = \frac{\operatorname{Tr}(H_t^\star) - \sum_{i \in \mathcal{K}} \widehat{\lambda}_i}{d - |\mathcal{K}|}. \tag{56}$$

Candidates satisfying $\widehat{\lambda}_i \le \tau_t$ are removed because they do not represent anisotropy above the resulting isotropic floor. After each removal, τ_t is recomputed using Eq. (56). The retained set can therefore contain fewer than R_H directions, and removed positions are not filled by lower-priority candidates.

The retained basis and spectrum are

$$U_t = [\,\widehat{u}_i\,]_{i\in\mathcal{K}}, \qquad \lambda_t = [\,\widehat{\lambda}_i\,]_{i\in\mathcal{K}}. \tag{57}$$

They define the bounded historical operator

$$H_t = U_t \,\mathrm{diag}\,(\lambda_t - \tau_t \mathbf{1})\, U_t^\top + \tau_t I. \tag{58}$$

Direct substitution gives

$$\mathrm{Tr}(H_t) = \mathrm{Tr}(H_t^\star), \tag{59}$$

which gives the trace-preservation property stated in Section 4.3.

A.6 Attenuation Properties

For $H \succeq 0$ with nonzero trace, RG-LoRA defines

$$S(H) = \left(I + H/(\mathrm{Tr}(H)/d)\right)^{-1/2}. \tag{60}$$

Its eigenvalues lie in $(0, 1]$, so $S(H)$ is invertible.

Matrix-free application. Write the stored history as $H = U\,\mathrm{diag}(\lambda - \tau\mathbf{1})U^\top + \tau I$ with $U^\top U = I$, and let $\bar{h} = \mathrm{Tr}(H)/d$. Define $g_0 = (1 + \tau/\bar{h})^{-1/2}$ and $g_i = (1 + \lambda_i/\bar{h})^{-1/2}$. Spectral calculus gives

$$S(H)X = g_0 X + U\,\mathrm{diag}(g_i - g_0)(U^\top X), \qquad XS(H) = g_0 X + (XU)\,\mathrm{diag}(g_i - g_0)U^\top. \tag{61}$$

For zero-trace PSD history, set $S(H) = I$. Thus applying the output map to $B \in \mathbb{R}^{d_o\times r}$ or the input map to $A \in \mathbb{R}^{r\times d_i}$ uses only at most R_H stored columns and costs $O(d_o r R_H)$ or $O(d_i r R_H)$, respectively. No dense history or attenuation matrix is materialized.

Proposition 5 (Scale and rank invariance)**.** *For every scalar $c > 0$, $S(cH) = S(H)$. Moreover, if L and T are RG-LoRA attenuation maps, then*

$$\{L(BA)T : \mathrm{rank}(BA) \le r\} = \{D : \mathrm{rank}(D) \le r\}. \tag{62}$$

Proof. The scalar cancels between cH and its mean trace, proving scale invariance. Both attenuation maps are invertible, so left and right multiplication preserve matrix rank. Conversely, every rank-r matrix D can be written as $L(L^{-1}DT^{-1})T$, and the middle matrix still has rank at most r. □

The identity-Jacobian surrogate $\mathrm{IS}(X, \widetilde{X}) = X + \mathrm{stopgrad}(\widetilde{X} - X)$ has forward value $\widetilde{X}$ and derivative I with respect to X. Using the same surrogate for both LoRA factors keeps the local Jacobian with respect to the raw factors equal to the identity, so the fixed attenuation maps do not add factor-space preconditioning through the chain rule.

Proposition 6 (Contraction of historical exposure)**.** *For fixed PSD histories $H^{\mathrm{out}}, H^{\mathrm{in}}$ with positive traces, let $L = S(H^{\mathrm{out}})$ and $T = S(H^{\mathrm{in}})$. For every candidate residual D,*

$$\mathcal{E}_H(LDT) \le \mathcal{E}_H(D), \qquad \mathcal{E}_H(D) = \mathrm{Tr}(H^{\mathrm{out}} D H^{\mathrm{in}} D^\top). \tag{63}$$

Proof. Write $H^{\mathrm{out}} = E_o\,\mathrm{diag}(\mu)E_o^\top$ and $H^{\mathrm{in}} = E_i\,\mathrm{diag}(\nu)E_i^\top$, with nonnegative spectra, and set $\widehat{D} = E_o^\top D E_i$. Each term $\mu_a\nu_b|\widehat{D}_{ab}|^2$ is multiplied by

$$\frac{1}{(1 + \mu_a/\bar{h}_o)(1 + \nu_b/\bar{h}_i)} \le 1, \qquad \bar{h}_o = \mathrm{Tr}(H^{\mathrm{out}})/d_o, \quad \bar{h}_i = \mathrm{Tr}(H^{\mathrm{in}})/d_i. \tag{64}$$

Summing proves the inequality. If either history is zero, its attenuation map is defined as the identity and the exposure is zero. □

This result compares a fixed residual with its attenuated version and states the directional effect of the task-fixed maps.

Table 5: Canonical configuration used for the main RG-LoRA experiments. Ablation-specific changes are stated separately.

Component	Value
LoRA rank / alpha	$r = 8$ / $\alpha_{\mathrm{LoRA}} = 16$
Adapted modules (T5)	q, v in encoder and decoder self-attention
	q, v in decoder cross-attention
Adapted modules (Llama)	q_proj, v_proj in all self-attention layers
History budget	$R_H = 64$ explicit directions per layer side
Per-task candidate rank	at most 8, inherited from the executed LoRA residual
Historical representation	one basis and spectrum per side plus scalar isotropic remainder τI
Historical target	equal-task consolidation target
Mode score	spectral value $\times$ bounded residual activity$^{0.25}$
Attenuation	two-sided, soft, trace-mean normalized, task-fixed
Task-end merge	direct addition to adapted backbone
Persistent task-specific state	none
Replay / task ID at inference	none / none

B Complete Algorithm and State Accounting

For each task and adapted layer, RG-LoRA performs the following operations:

1. construct task-fixed attenuation maps L_t and T_t from the bounded input- and output-side histories;
2. train one fresh zero-function rank-r LoRA on current-task data;
3. form the executed residual $\widetilde{D}_t = sL_tB_tA_tT_t$ and merge it directly into the adapted backbone;
4. extract the compact eigensystems of $\widetilde{D}_t^\top \widetilde{D}_t$ and $\widetilde{D}_t \widetilde{D}_t^\top$;
5. update both equal-task historical sides and compress each to at most R_H explicit directions with isotropic trace closure;
6. discard the active LoRA factors and temporary residual geometry.

For one adapted layer, persistent historical vector storage is

$$O(R_H(d_i + d_o)), \tag{65}$$

independent of the number of tasks. In addition to the adapted backbone, each layer stores two bounded bases and spectra, two scalar isotropic trace remainders, and the task counter. No previous-task examples or task-indexed adapters are retained.

Compared with ordinary LoRA, RG-LoRA adds left/right attenuation during training, a compact QR/SVD decomposition of the executed residual, and bounded consolidation of at most $R_H + r$ candidate directions per side at each task boundary. Inference uses the merged backbone and requires neither historical geometry nor a task identifier.

C Experimental Details

C.1 Canonical RG-LoRA Configuration

C.2 Prompt and Data Protocol

The decoder-only CL classification prompt is:

```
{definition}
Options: {label_1}, ..., {label_K}
{instance}
Output:
```

Table 6: Training and evaluation protocol. Short denotes CL O1–O3 and long denotes CL O4–O6.

Setting	T5-Large CL	Llama CL	Llama SuperNI
Epochs per task	1	1	5
Train batch / accumulation	4 / 2	2 / 8	2 / 4
Learning rate	2×10^{-4}	short: 2×10^{-5}, long: 10^{-4}	2×10^{-5}
Optimizer / schedule	AdamW / linear warmup	AdamW / constant	AdamW / constant
Weight decay / warmup	0.01 / 0.06	0 / 0	0 / 0
Maximum source / target length	512 / 50	1024 / 50	1024 / 50
Precision	BF16 compute, FP32 adapted weights	BF16 compute, FP32 LoRA/weights	BF16 compute, FP32 LoRA/weights
Gradient checkpointing	model default	enabled	enabled
Evaluation	final-only	final-only	final-only
Training examples	short: full, long: 1,000/task	short: full, long: 1,000/task	up to 1,000/task
Evaluation examples	short: full, long: up to 500/class	short: full, long: up to 500/class	up to 100/task
Main-result seeds	41, 42, 43	41, 42, 43	41, 42, 43

Empty definition or option fields omit the corresponding line.

The encoder–decoder CL prompt is:

```
Task:{group}
Dataset:{dataset}
{instruction}
Option: {label_1}, ..., {label_K}
{instance}
Answer:
```

For COPA, the instruction and option lines are omitted.

The SuperNI prompt is:

```
Definition: {definition}

Now complete the following example -
Input: {instance}
Output:
```

Decoder-only experiments use no native chat template, and only target tokens contribute to the language-model loss.

CL O1–O3 use all valid training and evaluation records. For O4–O6, records with missing or invalid labels are removed. Each task then uses at most 1,000 training examples sampled uniformly without replacement and up to 500 evaluation examples per class. T5-Large and Llama-2-7B-Chat use fixed task-level samples across seeds 41, 42, and 43. Llama-3.2-3B instead resamples the capped CL splits using the corresponding run seed, so its multi-seed variation also includes data selection. SuperNI uses up to 1,000 training and 100 evaluation examples per task.

For Llama CL, O1–O3 use a learning rate of 2×10^{-5} and O4–O6 use 10^{-4}. The long-sequence setting uses capped training sets and therefore fewer optimization steps per task. These learning rates are fixed for the corresponding task orders.

For decoder-only CL and SuperNI classification, predictions and references are lowercased, ASCII punctuation is removed, and whitespace is collapsed before exact comparison. A prediction is correct if it matches any valid reference. T5 CL uses the same exact-match procedure after lowercasing predictions and labels, stripping outer whitespace, replacing underscores and hyphens with spaces, and collapsing whitespace. SuperNI generation uses the maximum ROUGE-L Lin (2004) F-measure over the available references, with stemming.

C.3 Retention Metrics

Let $a_{i,j}$ denote the score on task j after completing task i. For a stream of T tasks, backward transfer is

$$\mathrm{BWT} = \frac{1}{T-1} \sum_{j=1}^{T-1} \left(a_{T,j} - a_{j,j} \right). \tag{66}$$

Positive BWT indicates improvement on earlier tasks after subsequent learning, while negative values indicate degradation.

Average forgetting is

$$\mathrm{Fgt} = \frac{1}{T-1} \sum_{j=1}^{T-1} \left(\max_{i \in \{j,\dots,T\}} a_{i,j} - a_{T,j} \right). \tag{67}$$

Lower forgetting is better. The final task is excluded because no subsequent task can affect it. For SuperNI, $a_{i,j}$ uses accuracy for classification tasks and ROUGE-L for generation tasks.

C.4 First-Step LoRA Initialization

All runs use zero-function LoRA initialization with $B = 0$. Before the first optimizer step, the first accumulation group provides the pre-attenuation layer gradient

$$G = \sum_{b,p} g_{b,p} x_{b,p}^{\top}, \tag{68}$$

where x is the LoRA-dropout output and g is the gradient at the adapted layer output.

For each layer, a randomized SVD of GT_t uses sketch size $q = \min(d_o, d_i, 2r)$ and two power iterations to obtain the leading r right singular vectors as rows of $V_r^{\top}$. The LoRA factor A is initialized as

$$A \leftarrow a_0 V_r^{\top}, \qquad a_0 = \frac{\|A_{\mathrm{init}}\|_F}{\sqrt{r}}, \tag{69}$$

while B remains zero, leaving the initial function unchanged.

The first-step factor gradients are

$$\nabla_B = sG(AT_t)^{\top}, \qquad \nabla_A = 0. \tag{70}$$

Standard gradient clipping and the optimizer update are then applied. This initialization reuses the first training backward group and requires no additional data pass or persistent state.

D Datasets, Outputs, and Task Orders

D.1 CL Benchmark Task Catalog

The DBpedia, AGNews, Yahoo, Yelp, and Amazon text-classification splits follow Zhang et al. (2015). IMDb and SST-2 follow Maas et al. (2011) and Socher et al. (2013), respectively. MNLI follows MultiNLI Williams et al. (2018). WiC Pilehvar & Camacho-Collados (2019), BoolQ Clark et al. (2019), and MultiRC Khashabi et al. (2018) provide the lexical-semantic and reading-comprehension tasks. RTE and QQP follow GLUE Wang et al. (2019b), while CB and COPA follow SuperGLUE Wang et al. (2019a). The continual subset, order, and generative label format follow the CL protocol described above, not the original benchmark leaderboards.

For completeness, DBpedia uses the labels *Company*, *Educational Institution*, *Artist*, *Athlete*, *Office Holder*, *Mean of Transportation*, *Building*, *Natural Place*, *Village*, *Animal*, *Plant*, *Album*, *Film*, and *Written Work*. Yahoo uses *Society & Culture*, *Science & Mathematics*, *Health*, *Education & Reference*, *Computers & Internet*, *Sports*, *Business & Finance*, *Entertainment & Music*, *Family & Relationships*, and *Politics & Government*. These strings are also the admissible normalized outputs used by exact-match evaluation.

D.2 Super-NaturalInstructions Task Catalog and Orders

SuperNI Wang et al. (2022) contains 1,616 instruction-defined NLP tasks; we follow the experimental setting of Merge Before Forget Qiao & Mahdavi (2026) for the fifteen-task subset and its two stream orders (their Tables 16–17), using up to 1,000 training and 100 test examples per task. Tables 9 and 10 reproduce the task selection and orders explicitly.

Table 7: All datasets appearing in CL orders O1–O6. Accuracy is used for every task after normalized exact matching to the listed output space.

Dataset	Family	Prediction problem	Output space
MNLI	NLI	three-way premise–hypothesis relation	neutral, entailment, contradiction
CB	NLI	commitment/factuality inference	entailment, contradiction, neutral
RTE	NLI	binary textual entailment	contradiction, entailment
WiC	lexical semantics	same word sense across two sentences	True, False
COPA	causal reasoning	choose the more plausible cause/effect	A, B
QQP	paraphrase	decide whether two questions have the same meaning	False, True
BoolQ	reading QA	passage-grounded yes/no question	True, False
MultiRC	reading QA	validate a candidate answer from a passage	False, True
IMDb	sentiment	long movie-review polarity	Good, Bad
Yelp	sentiment	five-way service-review sentiment	very negative, negative, neutral, positive, very positive
Amazon	sentiment	five-way product-review sentiment	very negative, negative, neutral, positive, very positive
SST-2	sentiment	short movie-text polarity	Good, Bad
DBpedia	topic/entity	classify an entity description	14 ontology classes (listed below)
AGNews	topic	classify a news article	World, Sports, Business, Sci/Tech
Yahoo	topic	classify a community question/answer	10 knowledge domains (listed below)

Table 8: Exact CL task orders. O1–O3 contain four tasks; O4–O6 contain all fifteen tasks.

Order	Task sequence
O1	DBpedia → Amazon → Yahoo → AGNews
O2	DBpedia → Amazon → AGNews → Yahoo
O3	Yahoo → Amazon → AGNews → DBpedia
O4	MNLI → CB → WiC → COPA → QQP → BoolQ → RTE → IMDb → Yelp → Amazon → SST-2 → DBpedia → AGNews → MultiRC → Yahoo
O5	MultiRC → BoolQ → WiC → MNLI → CB → COPA → QQP → RTE → IMDb → SST-2 → DBpedia → AGNews → Yelp → Amazon → Yahoo
O6	Yelp → Amazon → MNLI → CB → COPA → QQP → RTE → IMDb → SST-2 → DBpedia → AGNews → Yahoo → MultiRC → BoolQ → WiC

E Complete Results and Additional Ablations

E.1 Three-Seed Final-Score Variability

Table 13 gives each seed's final score, together with the across-seed mean and sample standard deviation. SC, LS, and SuperNI aggregates are computed within each seed first; their standard deviation is therefore not the average of the per-order standard deviations. SuperNI uses task-native scores. These final scores are not estimates of BWT or forgetting.

E.2 Published ASO-LoRA Protocol

ASO-LoRA Wang et al. (2026) reports the same six task orders used here (its Table 5). Its T5-Large results are taken from Table 1, its Llama O1–O3 results from Table 2, and its Llama O4–O6 results from Table 7. All order averages are kept as published. No SuperNI results are reported, so those cells remain unavailable rather than being inferred.

The source identifies the decoder-only backbone as Llama-2-7B in its implementation section, but does not specify whether it is the Chat variant. These numbers are therefore grouped by model family and kept separate from the Llama-2-7B-Chat runs reported here. Its reported training protocol uses one epoch, AdamW, batch size 64, learning rate 10^{-3}, zero weight decay, dropout 0.1, and averages over three runs. Our configuration is given in Table 6. We compare with ASO-LoRA's published results under its reported training protocol, while examining the contribution of soft attenuation separately through our ablations. ASO-LoRA also retains task-specific LoRA blocks, unlike RG-LoRA's bounded residual-geometry history and direct backbone merging.

E.3 Output-Side Attenuation Analysis

The attenuation sides act on different axes: DT controls the input directions on which a residual acts, whereas LD controls its layer-output directions. Shared singular values do not imply equal protection value or equal effects on learning. In particular, an output feature can serve both histori-

Table 9: The fifteen SuperNI tasks, following the task selection in Merge Before Forget Qiao & Mahdavi (2026).

ID	Short name	Task type	Metric
1572	SAMSum	dialogue summarization	ROUGE-L
363	SST-2 polarity	binary sentiment classification	accuracy
1290	XSum	news summarization	ROUGE-L
181	Outcome extraction	extract clinical-study outcomes	ROUGE-L
002	Quoref	reading-comprehension answer generation	ROUGE-L
1510	Relation extraction	generate the specified relation	ROUGE-L
639	MultiWOZ	user-utterance generation	ROUGE-L
1729	PersonaChat	next-response generation	ROUGE-L
073	CommonsenseQA	commonsense answer generation	ROUGE-L
1590	Diplomacy	dialogue continuation	ROUGE-L
748	GLUCOSE (reverse cause–event)	cause/event response generation	ROUGE-L
511	Reddit TIFU	long-text summarization	ROUGE-L
591	SciQ	science answer generation	ROUGE-L
1687	Sentiment140	sentiment classification	accuracy
875	Emotion	emotion classification	accuracy

Table 10: SuperNI task orders following Merge Before Forget Qiao & Mahdavi (2026). Numbers refer to the task IDs in Table 9.

Order	Task-ID sequence
O1	1572 → 363 → 1290 → 181 → 002 → 1510 → 639 → 1729 → 073 → 1590 → 748 → 511 → 591 → 1687 → 875
O2	748 → 073 → 1590 → 639 → 1572 → 1687 → 591 → 363 → 1510 → 1729 → 181 → 511 → 002 → 1290 → 875

cally used and novel input features. Output-only attenuation cannot distinguish those inputs merely because they produce the same output direction.

For a concrete fixed-candidate example, let $d_i = d_o = 2$, let u_1, u_2 be orthonormal inputs, and let v be a unit output direction. Take $H_i = u_1 u_1^\top$ and $H_o = vv^\top$, with zero isotropic floors. Both represented modes receive gain $g = 1/\sqrt{3}$. For $D = avu_1^\top + bvu_2^\top$, the three executed candidates are

$$\begin{aligned} DT &= gavu_1^\top + bvu_2^\top, \\ LD &= gavu_1^\top + gbvu_2^\top, \\ LDT &= g^2 avu_1^\top + gbvu_2^\top. \end{aligned} \tag{71}$$

Input-side attenuation reduces the historically represented input while leaving the novel input component unchanged. Output-side attenuation reduces both because they share v. Adding L further attenuates the old-input component, but it also attenuates the novel component. This example shows a possible protection–reuse trade-off; it does not show that this mechanism caused the observed degradation. Neither exposure contraction nor rank preservation guarantees monotone prediction gains from adding an attenuation side.

The four conditions in Table 4 use the same optimizer, data order, initialization protocol, training schedule, and evaluation procedure. They differ only in whether input-side attenuation, output-side attenuation, both, or neither are enabled. The comparison isolates the empirical effect of the two task-fixed attenuation maps under the reported protocol.

E.4 History-Budget Sensitivity

Increasing R_H from 64 to 128 gives a consistent but modest improvement on the three T5 long-sequence orders. The canonical setting remains $R_H = 64$ because it is used across all reported backbones and requires a smaller persistent history. The $R_H = 128$ result is reported only as a capacity sensitivity for T5.

E.5 Decoder-Only Rank-Sweep Protocol

Tables 15 and 16 use Llama-2-7B-Chat on CL O1, CL O4, and SuperNI O1, covering a four-task classification stream, a fifteen-task classification stream, and a fifteen-task heterogeneous instruction stream. Both sweeps use seed 42 and the canonical prompt, sampling, optimizer, and task-fixed

Table 11: Complete seed-42 T5-Large comparison (%). ASO-LoRA results are from ASO-LoRA Wang et al. (2026), Table 1; other published baselines follow ELLA Das Biswas et al. (2026).

	Standard CL				Long Sequence			
Method	O1	O2	O3	Avg.	O4	O5	O6	Avg.
SeqLoRA	39.5	31.9	46.6	39.3	4.9	3.5	4.2	4.2
EWC	46.3	45.3	52.1	47.9	44.9	44.0	45.4	44.8
LwF	52.7	52.9	48.4	51.3	49.7	42.8	46.9	46.5
LB-CL	76.9	76.5	76.8	76.7	68.4	67.3	71.8	69.2
O-LoRA	73.5	71.4	70.0	71.6	65.4	65.2	65.2	65.3
O-LoRA+MIGU	77.1	77.0	75.6	76.6	67.3	68.5	74.0	70.0
ELLA	80.0	80.0	79.8	79.9	73.4	72.0	75.4	73.6
ASO-LoRA	76.2	77.5	77.3	77.0	73.7	67.2	77.4	72.8
RG-LoRA	**80.20**	**80.10**	**80.65**	**80.32**	71.17	70.89	75.26	72.44

Table 12: Complete seed-42 Llama-2-7B-family comparison (%). ASO-LoRA results are from ASO-LoRA Wang et al. (2026), Tables 2 and 7 (Chat variant unspecified); other rows use Llama-2-7B-Chat and published baselines follow Merge Before Forget Qiao & Mahdavi (2026). Methods using retained or generated previous-task samples are omitted.

	Standard CL				Long Sequence				SuperNI		
Method	O1	O2	O3	Avg.	O4	O5	O6	Avg.	O1	O2	Avg.
SeqLoRA	73.3	76.2	78.4	76.0	69.1	66.0	71.1	68.7	18.4	26.8	22.6
O-LoRA	76.1	76.3	79.2	77.2	74.0	72.0	74.6	73.5	23.3	28.4	25.9
CorDA	78.4	79.3	80.0	79.2	73.4	72.7	74.0	73.4	20.9	16.0	18.5
MagMax	80.1	80.6	80.3	80.3	72.3	73.5	74.5	73.4	15.3	7.0	11.2
KnOTS	67.9	65.9	70.8	68.2	61.5	60.1	58.0	59.9	34.6	30.1	32.4
LoRA-LEGO	68.3	66.0	70.9	68.4	58.8	58.7	53.2	56.9	32.8	26.7	29.8
Merge Before Forget	80.1	80.8	80.4	80.4	75.0	74.4	75.1	74.8	38.7	35.7	37.2
ASO-LoRA	75.0	74.8	78.1	76.0	59.1	64.5	59.2	60.9	–	–	–
RG-LoRA	**81.33**	**81.11**	**80.52**	**80.99**	**80.13**	**81.85**	**81.71**	**81.23**	**52.03**	**50.92**	**51.48**

attenuation protocol. The learning rates are 2×10^{-5} for CL O1 and SuperNI O1 and 10^{-4} for CL O4, with one epoch per CL task and five epochs per SuperNI task. Remaining settings follow Table 6.

The history-budget sweep fixes $r = 8$ and $\alpha_{\mathrm{LoRA}} = 16$, changing only $R_H \in \{16, 32, 64, 128\}$ on each historical side. The LoRA-rank sweep fixes $R_H = 64$ and changes $r \in \{4, 8, 16\}$, setting $\alpha_{\mathrm{LoRA}} \in \{8, 16, 32\}$ so that the forward scale remains exactly two. Each task contributes its exact compact residual geometry of rank at most r; no separate local-rank truncation or residual vector budget is introduced. The sweeps comprise twelve and nine complete runs, respectively. The two sweeps were run independently. Their shared $(r, R_H) = (8, 64)$ setting is a separate execution in each table, and each saved result is reported as produced by that sweep. Since evaluation is final-only, these tables report final average performance, not backward transfer or forgetting.

F Implementation Invariants

A correct RG-LoRA implementation checks the following conditions at every task boundary:

1. **Executed-residual parity:** the matrix used to construct history equals the update merged into the backbone, including LoRA scaling and both attenuation maps.
2. **Dense-Gram parity:** on small random layers, compact eigenpairs reconstruct $\widetilde{D}_t^\top \widetilde{D}_t$ and $\widetilde{D}_t \widetilde{D}_t^\top$ to numerical tolerance.
3. **Residual trace equality:** $\mathrm{Tr}(G_t^{\mathrm{in}}) = \mathrm{Tr}(G_t^{\mathrm{out}}) = \|\widetilde{D}_t\|_F^2$.
4. **PSD and orthogonality:** spectra and τ are nonnegative, and every stored basis obeys $\|U^\top U - I\|_\infty \le \varepsilon_{\mathrm{orth}}$.
5. **Rank bound:** neither historical side exceeds R_H explicit vectors, and no hidden vector store expands that budget.

Table 13: Complete RG-LoRA final scores across seeds 41, 42, and 43 (%). Mean and SD use the unrounded scores; SD is the sample standard deviation. SC/LS aggregate O1–O3/O4–O6, and SNI aggregates its two orders.

Backbone	Order	Seed 41	Seed 42	Seed 43	Mean	SD
T5-Large	O1	79.56	80.20	79.98	79.91	0.33
	O2	79.99	80.10	79.70	79.93	0.20
	O3	78.82	80.65	78.33	79.27	1.22
	SC avg.	79.46	80.32	79.34	**79.70**	0.53
	O4	70.21	71.17	71.44	70.94	0.65
	O5	75.60	70.89	74.71	73.73	2.51
	O6	73.57	75.26	75.56	74.80	1.07
	LS avg.	73.13	72.44	73.90	**73.16**	0.73
Llama-2-7B-Chat	O1	80.67	81.33	81.36	81.12	0.39
	O2	81.03	81.11	81.24	81.13	0.10
	O3	80.65	80.52	81.31	80.83	0.42
	SC avg.	80.79	80.99	81.30	**81.03**	0.26
	O4	81.44	80.13	82.51	81.36	1.19
	O5	81.16	81.85	81.79	81.60	0.39
	O6	81.08	81.71	82.12	81.64	0.52
	LS avg.	81.23	81.23	82.14	**81.53**	0.53
	SNI-1	52.05	52.03	52.77	52.29	0.42
	SNI-2	51.77	50.92	52.24	51.64	0.67
	SNI avg.	51.91	51.48	52.50	**51.96**	0.52
Llama-3.2-3B	O1	81.61	81.39	81.49	81.50	0.11
	O2	81.79	81.26	81.51	81.52	0.26
	O3	80.76	80.47	81.38	80.87	0.47
	SC avg.	81.38	81.04	81.46	**81.29**	0.22
	O4	79.91	79.70	79.78	79.80	0.10
	O5	80.30	80.99	79.71	80.33	0.64
	O6	80.70	80.74	80.34	80.59	0.22
	LS avg.	80.30	80.48	79.94	**80.24**	0.27
	SNI-1	49.95	50.87	49.79	50.20	0.58
	SNI-2	51.87	52.48	53.81	52.72	0.99
	SNI avg.	50.91	51.68	51.80	**51.46**	0.48

Table 14: T5-Large long-sequence sensitivity to the history budget (%).

R_H	O4	O5	O6	Avg.
64	71.17	70.89	75.26	72.44
128	**72.50**	**71.33**	**75.72**	**73.18**

6. **Trace closure:** relative trace error after compression is at most the declared numerical tolerance.
7. **Scale invariance:** multiplying a history side by a positive scalar does not alter its task-fixed attenuation map.
8. **Boundary-merge parity:** the backbone weight difference across a boundary matches $\widetilde{D}_t$.
9. **Transient-state deletion:** active LoRA factors and temporary residual geometry are absent after consolidation.

These checks separate numerical stabilization from changes to the method. Higher-precision QR/SVD or basis canonicalization may correct floating-point drift, but they must preserve the represented trace and the executed residual.

Table 15: History-budget sensitivity (%) on Llama-2-7B-Chat; LoRA $r = 8$.

R_H	CL O1	CL O4	SNI O1
16	81.09	79.91	46.93
32	81.34	80.06	49.01
64	**81.43**	80.22	51.76
128	81.36	**80.54**	**52.69**

Table 16: LoRA-rank sensitivity (%) on Llama-2-7B-Chat; $R_H = 64$ and $\alpha_{\mathrm{LoRA}}/r = 2$.

r	CL O1	CL O4	SNI O1
4	81.13	80.35	**53.09**
8	81.43	80.11	52.28
16	**81.75**	**80.61**	50.97