Provable Forgetting Bounds Drive Capacity Savings: Spectral Thresholding in Continual LoRA

Anonymous Authors[1]

Abstract

Orthogonal-subspace LoRA methods mitigate forgetting in continual learning of foundation models by assigning each task a new adapter subspace, but existing approaches typically allocate a fixed, hand-picked rank to every task and layer. This ignores a key source of heterogeneity: the amount of rank needed to avoid interference depends on the residual spectrum of the task's layerwise activations. We introduce DyRA, a theory-guided dynamic rank allocation rule that selects adapter bases by retaining residual singular directions whose singular value exceeds a global threshold τ. Our method is motivated by a per-layer interference bound showing that, for orthogonal LoRA, future-task interference on a past task is controlled by the spectral tail left outside the past-task adapter basis. Thus, DyRA replaces a fixed-rank design choice with a spectral-tail control principle. On continual instruction tuning of the LLaMA-2 7B foundation model with TRACE, DyRA yields a favorable performance–capacity trade-off, with the clearest gains at low rank ($+4.7$ AP over fixed $r{=}4$) and matches fixed $r{=}8$ with about 31% fewer adapter parameters. DyRA requires only one residual-spectrum computation per task and layer, and can be added to existing orthogonal-subspace LoRA methods.

1. Introduction

Low-rank adapters (LoRA) (Hu et al., 2022) make it tractable to fine-tune a large pretrained model, but in continual learning each new adapter also *perturbs* the outputs the model would have produced on past tasks. Orthogonal-subspace variants such as InfLoRA (Liang & Li, 2024) mitigate forgetting by forcing each new adapter to act only in directions orthogonal to past-task activations. Their orthogonality constraint is, however, *structural*: it is imposed on a rank-limited basis of past-task activations, and comes with no quantitative bound on how much the new task still perturbs past-task outputs through the *uncaptured* directions. A second, independent limitation is that the adapter rank r is chosen as a *single global hyperparameter* shared across all tasks and all layers. This is wasteful in both directions: layers and tasks with a rapidly decaying residual spectrum need far less capacity, while others are starved of it.

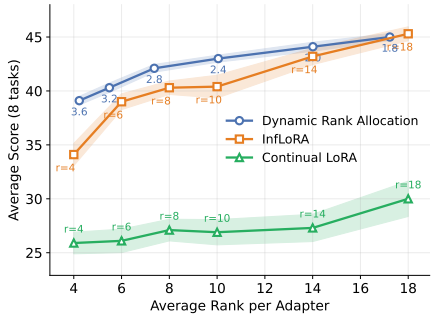

Figure 1. **Performance–capacity trade-off.** Average TRACE score (8 tasks, LLaMA-2 7B) vs. average rank per adapter, for dynamic rank via a spectral threshold $\tau \in \{1.8, \ldots, 3.6\}$ (ours, blue) and fixed-rank orthogonal-subspace LoRA. Full numerical values in Table 3.

Contributions. We take a theory-first route to the design of Continual Adapters methods, *with guarantees on forgetting.* In doing so, we develop a cheap, simple to implement and effective dynamic rank allocation version of InfLora. *(i)* Starting from a layerwise no-interference condition, we derive a generic end-to-end drift bound for continual orthogonal LoRA in which each adapted layer appears through a single per-layer tolerance ε_k^ℓ (Section 2). *(ii)* We show that the residual-subspace construction popularized by InfLoRA (Liang & Li, 2024) *realizes* this tolerance as the Frobenius norm of the uncaptured spectral tail of the task's layerwise residual activations (Lemma 3.1, Section 3). *(iii)* Under this realization, rank selection reduces—via Cauchy–Schwarz—to a Lipschitz-free, separable squared-tail surrogate whose optimal solution is a single-threshold rule on the

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the ICML 2026 Workshop "Connecting Low-rank Representations in AI" (CoLoRAI). Do not distribute.

residual singular values: retain directions with $\sigma^{\ell}_{k,j,\text{res}} \geq \tau$ (Section 4). *(iv)* On LLaMA-2 7B continual instruction tuning with TRACE, the rule matches fixed $r{=}8$ at $\bar{r}{\approx}5.5$ ($\approx 31\%$ fewer parameters) and improves on fixed $r{=}4$ by ≈ 4.7 AP (Section 5 and Figure 1).

Related Work. *Orthogonal-subspace continual LoRA* (O-LoRA (Wang et al., 2023a), InfLoRA (Liang & Li, 2024), and KeepLoRA (Luo et al., 2026)) control forgetting through a structural orthogonality constraint at a fixed global rank, but provides no quantitative control over the interference left uncaptured by truncation. *Adaptive-rank LoRA* (AdaLoRA (Zhang et al., 2023b) and follow-ups (Liu et al., 2024; Chang et al., 2025; Ding et al., 2023; Zhang et al., 2023a; Valipour et al., 2023)) allocates rank from importance / sensitivity heuristics unrelated to forgetting. We instead derive an *end-to-end forgetting bound* for continual orthogonal LoRA (Equation (3)) in which the same residual-spectrum tail controls both the adapter basis (Lemma 3.1) and, via a Cauchy–Schwarz reduction, the optimal per-layer rank allocation (Equation (7)). To our knowledge this is the first such bound in this family. Section A discusses additional prior work (gradient-projection, NTK-based forgetting analyses).

Preliminaries. We study continual instruction tuning of a pretrained transformer on a sequence of tasks $\{\mathcal{T}_t\}_{t=1}^{T}$ with datasets D_t arriving in order. We adopt the column-token convention: at layer ℓ, the input activation matrix is $X^{\ell} \in \mathbb{R}^{d\times n}$ (columns are tokens) - the superscript ℓ is omitted for input. LoRA freezes each linear layer at its pretrained weight W_0^{ℓ} and adds $\Delta W_t^{\ell} = A_t^{\ell} B_t^{\ell}$ with $A_t^{\ell} \in \mathbb{R}^{d_{\text{out}}\times r_t^{\ell}}$, $B_t^{\ell} \in \mathbb{R}^{r_t^{\ell}\times d}$ of rank $r_t^{\ell} \ll d$, so that the effective weight at layer ℓ after task t is $W_t^{\ell} := W_0^{\ell} + \sum_{i\leq t} \Delta W_i^{\ell}$. Write $\mathcal{A}_t := \{(A_i^{\ell}, B_i^{\ell})\}_{i\leq t,\,\ell}$ for the adapter parameters after task t, and $f_{\mathcal{A}_t}$ for the resulting full-network map. Let $x_k^{\ell}(s)$ denote the layer-ℓ input when processing task-k data under $\mathcal{A}_s$. We call $\|f_{\mathcal{A}_t}(X_k) - f_{\mathcal{A}_k}(X_k)\|_F$ the *forgetting* on task k at time $t \geq k$; the special case in which this quantity is 0 is *zero-forgetting*.

2. Spectral Interference Analysis

Layerwise no-interference. Let $k \leq t$ be a past task. A short induction on the computation graph (Section B) shows that zero-forgetting on task k holds as soon as, at every adapted layer ℓ,

$$W_t^{\ell}\, x_k^{\ell}(k) \;=\; W_k^{\ell}\, x_k^{\ell}(k). \tag{1}$$

While strict enforcement of (1) *does* yield zero forgetting, in practice it leads to very high per-task ranks and a rapid loss of plasticity (Section C).

Relaxation to a per-layer tolerance. We therefore introduce, for each layer ℓ and past task k, a scalar design budget $\varepsilon_k^{\ell} \geq 0$ and replace the equality (1) by

$$\left\|(W_t^{\ell} - W_k^{\ell})\, x_k^{\ell}(k)\right\|_F \;\leq\; \varepsilon_k^{\ell}. \tag{2}$$

At this point ε_k^{ℓ} is *purely generic*: any mechanism that produces such a tolerance feeds directly into the next steps.

End-to-end drift. Because transformer sub-layers are Lipschitz (layer norms, softmax, frozen MLPs), the per-layer tolerances compose (Theorems E.1 and E.2, proved in Section E) into an end-to-end bound on forgetting of the form

$$\begin{aligned}&\left\|f_{\mathcal{A}_t}(X_k) - f_{\mathcal{A}_k}(X_k)\right\|_F \\ &\quad\leq \sum_{i=0}^{L-1}\left(1 + L^i_{\text{MLP}} L^i_{\text{LN},2}\right) B^i_{\text{att}}(\varepsilon) \prod_{j>i} a_j,\end{aligned} \tag{3}$$

where $B^i_{\text{att}}(\varepsilon) = \sum_{\Pi\in\{Q,K,V\}} \alpha^i_{\Pi}\, \varepsilon_k^{\Pi,i}$ aggregates the three attention projections of block i through softmax-Lipschitz coefficients α^i_{Π}, and a_j collects Lipschitz factors of downstream sub-layers. The reading is: control every ε_k^{ℓ} and you control end-to-end forgetting. The next two sections ask, respectively, *how* the ε_k^{ℓ} are produced (Section 3) and *how the ranks producing them should be chosen* (Section 4).

3. Bounding Layerwise Interference

Residual-subspace construction. To realize a small ε_k^{ℓ} one needs $\|\Delta W_i^{\ell}\, x_k^{\ell}(k)\|$ to be small for every future task $i > k$. InfLoRA (Liang & Li, 2024) ensures this by restricting adapter inputs to past-orthogonal directions: for each task t and layer ℓ one builds a row-orthonormal basis $B_t^{\ell} \in \mathbb{R}^{r_t^{\ell}\times d}$ that is orthogonal to all previous bases, i.e., $B_i^{\ell}\,(B_j^{\ell})^{\top} = 0$ for $i \neq j$.

$$X^{\ell}_{t,\text{res}} \;:=\; P^{\perp}_{1:t-1}\, X_t^{\ell}, \qquad P^{\perp}_{1:t-1} := I - \sum_{i<t} (B_i^{\ell})^{\top} B_i^{\ell},$$

B_t^{ℓ} is chosen as the top-r_t^{ℓ} left singular vectors of $X^{\ell}_{t,\text{res}}$; only A_t^{ℓ} is then trained, with B_t^{ℓ} frozen. The original method uses a fixed rank $r_t^{\ell} = r$ across all tasks and layers.

Tail-norm realization of ε_k^{ℓ}. Let $X^{\ell}_{k,\text{tail}} := X^{\ell}_{k,\text{res}} - (B_k^{\ell})^{\top} B_k^{\ell}\, X^{\ell}_{k,\text{res}}$ be the uncaptured part of the residual activations.

Lemma 3.1 (Bounded layerwise interference under the residual-subspace construction)**.** *Under the construction above,*

$$\begin{aligned}&\left\|(W_t^{\ell} - W_k^{\ell})\, x_k^{\ell}(k)\right\|_F \\ &\quad\leq \Big(\sum_{i=k+1}^{t} \|A_i^{\ell}\|_2\Big)\, \|X^{\ell}_{k,\mathit{tail}}\|_F,\end{aligned} \tag{4}$$

for all $t \geq k$. Moreover, taking B_k^ℓ to be the top-r_k^ℓ left singular vectors of $X_{k,res}^\ell$ minimizes $\|X_{k,tail}^\ell\|_F$ over all rank-r_k^ℓ orthonormal matrices in the past-orthogonal complement (Eckart–Young–Mirsky (Mirsky, 1960)), and yields $\|X_{k,tail}^\ell\|_F^2 = \sum_{j>r_k^\ell} (\sigma_{k,j,res}^\ell)^2$.

Equation (4) realizes the generic per-layer tolerance of (2) by $\varepsilon_k^\ell := \left(\sum_{i>k} \|A_i^\ell\|_2\right) \|X_{k,\text{tail}}^\ell\|_F$: under InfLoRA's construction, ε_k^ℓ splits into a post-hoc term $\sum_{i>k} \|A_i^\ell\|_2$ (depending on future training) and the *tail norm* $\|X_{k,\text{tail}}^\ell\|_F$, a function of the residual spectrum and the chosen rank that we *can* control at time k. Substituting (4) into (3) yields an end-to-end forgetting bound whose only free parameter is the collection of ranks $\{r_k^\ell\}$. Section 4 asks: how should they be chosen?

4. Theory-Guided Rank Allocation

Substituting (4) into (3) bounds the end-to-end forgetting at task k by an ℓ^1-style sum across adapted layers,

$$\sum_\ell C_k^\ell \, T_\ell(r_k^\ell), \qquad T_\ell(r) := \Big(\sum_{j>r} (\sigma_{k,j,\text{res}}^\ell)^2\Big)^{1/2},$$

i.e. $T_\ell(r) = \|X_{k,\text{tail}}^\ell\|_F$ at rank r, with weight C_k^ℓ bundling the Lipschitz factors of (3) together with the post-hoc factor $\sum_{i>k} \|A_i^\ell\|_2$ of (4). The latter depends on adapters trained *after* task k and is therefore unavailable at allocation time, so we cannot directly minimize this bound.

Cauchy–Schwarz surrogate. Cauchy–Schwarz in ℓ^2 gives, after squaring,

$$\Big(\sum_\ell C_k^\ell \, T_\ell(r_k^\ell)\Big)^2 \;\leq\; \underbrace{\Big(\sum_\ell (C_k^\ell)^2\Big)}_{=:K_k,\ \text{indep. of } \{r_k^\ell\}} \cdot \sum_\ell T_\ell(r_k^\ell)^2. \tag{5}$$

The constant K_k does not depend on the ranks, so minimizing this Cauchy–Schwarz upper bound is equivalent to minimizing the squared-tail surrogate

$$\sum_\ell T_\ell(r_k^\ell)^2 \;=\; \sum_{\ell,\, j>r_k^\ell} (\sigma_{k,j,\text{res}}^\ell)^2.$$

Two consequences. *(a) Lipschitz-free.* The unknown weights C_k^ℓ have collapsed into the rank-independent constant K_k: the rank allocation problem no longer depends on the post-hoc adapter norms or the Lipschitz factors of (3). *(b) Separable across (ℓ, j).* The surrogate is a sum of squared singular values indexed by per-layer directions (ℓ, j), and each unit of rank assigned to (ℓ, j) removes exactly $(\sigma_{k,j,\text{res}}^\ell)^2$ from it. The CS step suppresses the per-layer asymmetry of C_k^ℓ in favor of K_k. Section G discusses the slack of this step and alternative-norm surrogates.

Rank allocation problem. Under a per-task adapter-rank budget $R_k = \sum_\ell r_k^\ell$ (equivalent to a parameter budget under uniform per-rank cost; see Section H), the Cauchy–Schwarz surrogate of (5) yields

$$\begin{aligned} &\min_{\{r_k^\ell\}} \sum_{\ell,\, j>r_k^\ell} (\sigma_{k,j,\text{res}}^\ell)^2 \\ &\text{s.t.} \sum_\ell r_k^\ell \leq R_k, \;\; r_k^\ell \in \{0, \ldots, \rho_k^\ell\}, \end{aligned} \tag{6}$$

with $\rho_k^\ell := \text{rank}(X_{k,\text{res}}^\ell)$.

Greedy allocation is singular-value thresholding. Because each unit of rank at (ℓ, j) removes exactly $(\sigma_{k,j,\text{res}}^\ell)^2$ from the objective of (6) and contributes a single unit to the budget, (6) is solved greedily by sorting all (ℓ, j) pairs by $\sigma_{k,j,\text{res}}^\ell$ and keeping the top-R_k. Equivalently, every such top-R_k set is the level set of a single global cutoff $\tau \geq 0$ (the dual variable of the budget constraint, the smallest selected singular value), and the optimal allocation is

$$r_k^\ell \;=\; r_{\tau,k}^\ell \;:=\; \#\big\{\, j : \sigma_{k,j,\text{res}}^\ell \geq \tau \,\big\}, \tag{7}$$

with τ in one-to-one correspondence with the budget R_k. By construction, every uncaptured singular value at layer ℓ lies below τ, yielding the per-layer tail bound $\|X_{k,\text{tail}}^\ell\|_F^2 \leq \tau^2(\rho_k^\ell - r_{\tau,k}^\ell)$ (Theorem F.1); lowering τ therefore shrinks every per-layer injection of (3), a uniform guarantee no fixed global rank provides.

Implementation. The threshold rule (7) is a single line of code on top of any orthogonal-subspace method. We accumulate the $d \times d$ Gram matrix $G = X_{k,\text{res}}(X_{k,\text{res}})^\top$ online during one forward pass (no activations stored), compute its eigendecomposition $G = U\Sigma^2 U^\top$, and retain the columns of U with $\sigma_i \geq \tau$. Cost is dominated by the forward pass; the eigendecomposition is $O(d^3)$ with $d \leq 4096$. In practice we sweep τ rather than R_k because the residual singular spectrum is empirically on a comparable scale across layers (see Figure 3), so a single τ yields meaningful layerwise allocations; in our experiments, $\tau \in \{1.8, 2.0, 2.4, 2.8, 3.2, 3.6\}$.

5. Preliminary Evidence

The theory of Sections 2 to 4 predicts three testable behaviors: (i) at matched capacity, thresholding dominates fixed rank; (ii) induced ranks should be heterogeneous, concentrated where the residual spectrum is rich; (iii) the forgetting (BWT) gap should favor the thresholded rule at comparable capacity. We now evaluate each prediction.

Setup. We continually fine-tune LLaMA-2 7B (Touvron et al., 2023) on the TRACE benchmark (Wang et al., 2023b)

(eight heterogeneous NLP tasks presented in a fixed order). Adapters are inserted at the W_Q and W_V projections of every self-attention layer (64 adapted projections, 32 transformer layers). For orthogonal-subspace baselines (InfLoRA and ours) only A_t is trained (5 epochs, AdamW, lr 10^{-4}); B_t is frozen after the SVD of $X_{t,\text{res}}$. All InfLoRA numbers are re-run by us under matched compute, hyperparameters, and seeds with our method. Let $R_{t,k}$ denote the test score on task k after training through task t. We report *average performance* $\text{AP} := \frac{1}{T}\sum_{k=1}^{T} R_{T,k}$ and *backward transfer* $\text{BWT} := \frac{1}{T-1}\sum_{k=1}^{T-1}(R_{T,k} - R_{k,k})$. See the supplement for full details.

(i) Trade-off: clean wins at low rank. Figure 1 plots AP as a function of the average rank per adapter. Thresholding ($\tau \in \{1.8, \ldots, 3.6\}$) is at least on par with fixed-rank InfLoRA ($r \in \{4, 6, 8, 10, 14, 18\}$) at every matched-capacity point. High-rank gaps lie within one seed std; the clear signal is at low rank, where fixed allocation leaves a large uncovered tail on spectrum-rich layers and (8) caps it uniformly. At τ=3.6 ($\bar{r} \approx 4.2$), AP is 38.7 ± 1.2 vs. 34.1 ± 1.8 for fixed r=4, a $\sim$ 4.7-point gap. At τ=3.2 ($\bar{r} \approx 5.5$), 41.2 ± 2.5 is comparable to fixed r=6 (39.95 ± 2.19) and to fixed r=8 (39.57 ± 4.45; Table 3), i.e., $\sim$8–31% fewer parameters at matched performance. For context, continual LoRA *without* any projection constraint saturates at $\approx$27% AP across the entire rank sweep—a $\sim$15-percentage-point gap below any orthogonal-subspace method—confirming that the headline gains come from the combination of orthogonality and dynamic allocation rather than from raw LoRA capacity.

(ii) Heterogeneous allocation. Figure 2 reports the rank selected per task at $\tau = 3.2$. Allocation varies by $> 3\times$ across tasks, with some receiving much less than fixed $r = 8$ and others substantially more—consistent with our thesis that a global r is generically sub-optimal. Figure 3 shows a clear depth-wise trend: selected ranks are near zero in early layers and grow with depth, with a smaller value at the last layer. Qualitatively similar patterns hold across τ and tasks.

(iii) Forgetting. At matched capacity, thresholding does not hurt BWT and tends to slightly improve it (Table 2 in Section I), consistent with (8): a smaller per-layer tail translates into a smaller per-layer interference and hence a smaller guaranteed end-to-end drift. With only T=8 tasks the BWT signal is modest; longer task sequences should amplify the gap.

6. Conclusion

A single-line change—pick the basis by a spectral threshold rather than a global rank—matches fixed-rank InfLoRA with up to $\sim$31% fewer parameters, as a direct consequence of Lemma 3.1. Because the same τ knob appears in every term of (3), it gives unified control over the plasticity–stability trade-off, with explicit guarantees on forgetting essential for safety-critical continual adaptation of foundation models. Tighter bounds (e.g. alternative norms; Section G), instance-dependent surrogates, and curriculum-style task ordering that minimizes total adapter capacity are natural next steps.

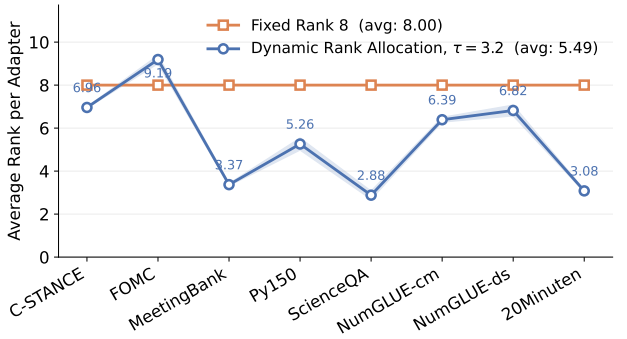

Figure 2. **Rank per task.** Average rank per task at $\tau = 3.2$ vs. fixed $r = 8$.

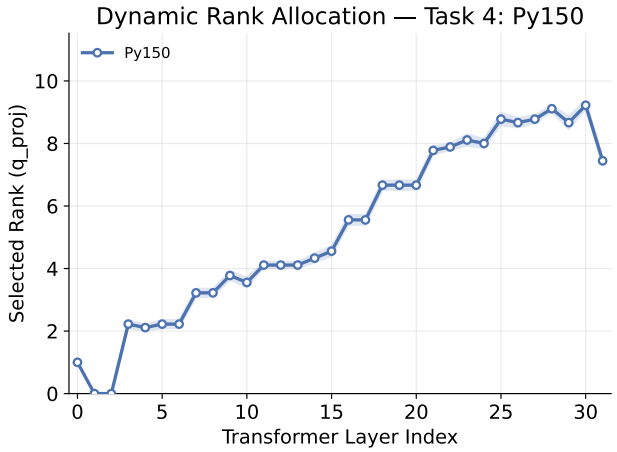

Figure 3. **Rank per layer.** Rank of `q_proj` adapters across transformer layers ($\tau = 2.8$, Py150).

Impact statement

By matching fixed-rank accuracy with $\sim$31% fewer adapter parameters, DyRA reduces the per-task memory and compute footprint of continually adapting foundation models, supporting more sustainable lifelong fine-tuning.

References

Chang, H., Ma, Z., Ma, M., Qi, Z., Sabot, A., Jiang, H., and Kung, H. T. ElaLoRA: Elastic & learnable low-rank adaptation for efficient model fine-tuning. *arXiv preprint arXiv:2504.00254*, 2025.

Ding, N., Lv, X., Wang, Q., Chen, Y., Zhou, B., Liu, Z., and Sun, M. Sparse low-rank adaptation of pre-trained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

Doan, T., Abbana Bennani, M., Mazoure, B., Rabusseau, G., and Alquier, P. A theoretical analysis of catastrophic forgetting through the NTK overlap matrix. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

Evron, I., Moroshko, E., Ward, R., Srebro, N., and Soudry, D. How catastrophic can catastrophic forgetting be in linear regression? In *Proceedings of the 35th Conference on Learning Theory (COLT)*, 2022.

Farajtabar, M., Azizan, N., Mott, A., and Li, A. Orthogonal gradient descent for continual learning. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Liang, Y.-S. and Li, W.-J. InfLoRA: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Lin, S., Ju, P., Liang, Y., and Shroff, N. Theory on forgetting and generalization of continual learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

Liu, Z., Lyn, J., Zhu, W., Tian, X., and Graham, Y. ALoRA: Allocating low-rank adaptation for fine-tuning large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2024.

Luo, M.-L., Zhou, Z.-H., Zhang, Y.-L., Wan, Y., Wei, T., and Zhang, M.-L. KeepLoRA: Continual learning with residual gradient adaptation. *arXiv preprint arXiv:2601.19659*, 2026.

Mirsky, L. Symmetric gauge functions and unitarily invariant norms. *The Quarterly Journal of Mathematics*, 11(1): 50–59, 1960.

Peng, L., Giampouras, P. V., and Vidal, R. The ideal continual learner: An agent that never forgets. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

Saha, G., Garg, I., and Roy, K. Gradient projection memory for continually learning neural networks. In *International Conference on Learning Representations (ICLR)*, 2021.

Touvron, H., Martin, L., Stone, K., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Valipour, M., Rezagholizadeh, M., Kobyzev, I., and Ghodsi, A. DyLoRA: Parameter-efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 2023.

Wang, X., Chen, T., Ge, Q., Xia, H., Bao, R., Zheng, R., Zhang, Q., Gui, T., and Huang, X. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023a.

Wang, X., Zhang, Y., Chen, T., Gao, S., Jin, S., Yang, X., Xi, Z., Zheng, R., Zou, Y., Gui, T., Zhang, Q., and Huang, X. TRACE: A comprehensive benchmark for continual learning in large language models. *arXiv preprint arXiv:2310.06762*, 2023b.

Zhang, F., Li, L., Chen, J., Jiang, Z., Wang, B., and Qian, Y. IncreLoRA: Incremental parameter allocation method for parameter-efficient fine-tuning. *arXiv preprint arXiv:2308.12043*, 2023a.

Zhang, Q., Chen, M., Bukharin, A., He, P., Cheng, Y., Chen, W., and Zhao, T. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2023b.

A. Extended related work

Forgetting guarantees in continual learning. Several continual-learning methods control forgetting through projection or subspace constraints. OGD (Farajtabar et al., 2020) projects new-task gradients to preserve previous-task outputs in a linearized regime, while GPM (Saha et al., 2021) stores activation subspaces and restricts future updates to their orthogonal complement. Other works derive forgetting characterizations in NTK or linear settings (Doan et al., 2021; Evron et al., 2022; Lin et al., 2023; Peng et al., 2023). These analyses share a no-interference motivation but operate at the gradient or NTK level, whereas our bound works inside the orthogonal LoRA architecture: it expresses past-task drift through a per-layer residual spectral tail, making forgetting explicitly rank-dependent.

Orthogonal-subspace continual LoRA. The closest PEFT methods reduce interference by enforcing orthogonality between task-specific adapter subspaces. O-LoRA (Wang et al., 2023a) imposes orthogonality in LoRA weight space, InfLoRA (Liang & Li, 2024) builds each adapter in the residual subspace of previous-task activations, and KeepLoRA (Luo et al., 2026) further constrains updates relative to pretrained and previous-task feature directions. These methods rely on a structural orthogonality mechanism, but still choose a single fixed rank r across tasks and layers. DyRA instead quantifies the remaining interference after truncation: Lemma 3.1 shows that the uncaptured residual spectral tail controls the layerwise drift, turning rank into a forgetting-control variable.

Adaptive-rank LoRA. Dynamic-rank LoRA methods allocate capacity non-uniformly across modules, typically using importance scores, gates, sensitivity estimates, or rank-growth schedules (Zhang et al., 2023b; Liu et al., 2024; Chang et al., 2025; Ding et al., 2023; Zhang et al., 2023a; Valipour et al., 2023). These methods improve parameter efficiency in standard fine-tuning, but their allocation criteria are not derived from a bound on past-task functional drift. Our allocation is different: it follows from minimizing a tail-norm surrogate of the forgetting bound, so the same spectral quantity both constructs the orthogonal adapter basis and determines how much rank each task and layer receives.

B. Layerwise no-interference implies zero-forgetting

We justify the claim used in (1) of Section 2: if, at every adapted layer ℓ, $W_t^\ell\, x_k^\ell(k) = W_k^\ell\, x_k^\ell(k)$, then $f_{\mathcal{A}_t}(X_k) = f_{\mathcal{A}_k}(X_k)$.

Proof sketch. We proceed by induction on the depth ℓ of the computation graph. The task-k inputs X_k enter the network identically under $\mathcal{A}_t$ and $\mathcal{A}_k$, and every sub-layer preceding the first adapted layer (frozen embeddings, frozen layer norms) is a deterministic function of its input; hence for the first adapted layer we have $x_k^\ell(t) = x_k^\ell(k)$, so the hypothesis directly yields $W_t^\ell\, x_k^\ell(t) = W_k^\ell\, x_k^\ell(k)$, i.e., identical outputs of layer ℓ under $\mathcal{A}_t$ and $\mathcal{A}_k$. All non-adapted sub-layers in between (layer norms, softmax, frozen MLPs, residual connections) are deterministic functions of their input and therefore propagate equality forward unchanged. Hence the input to the next adapted layer ℓ' also satisfies $x_k^{\ell'}(t) = x_k^{\ell'}(k)$, and the inductive step repeats. Reaching the output of the network gives $f_{\mathcal{A}_t}(X_k) = f_{\mathcal{A}_k}(X_k)$. □

C. Strict no-interference: zero forgetting, no plasticity

A concrete construction (extending InfLoRA). Strict no-interference can be realized by a one-line modification of InfLoRA, or equivalently by sending our threshold $\tau \to 0^+$. Let

$$X_{t,\mathrm{res}}^\ell \;=\; U_t^\ell\, \Sigma_t^\ell\, (V_t^\ell)^\top$$

be the compact SVD of the residual activations of task t at layer ℓ, with non-zero singular values $\sigma_{t,1,\mathrm{res}}^\ell \geq \cdots \geq \sigma_{t,r^\star,\mathrm{res}}^\ell > 0$ where $r^\star := \mathrm{rank}(X_{t,\mathrm{res}}^\ell)$. Whereas InfLoRA sets B_t^ℓ to the *top-r* left singular vectors with r a fixed hyperparameter, the strict-enforcement variant takes

$$B_t^\ell \;:=\; (U_t^\ell)^\top \in \mathbb{R}^{r^\star \times d}, \qquad r_t^\ell \;=\; r^\star \;=\; \mathrm{rank}(X_{t,\mathrm{res}}^\ell),$$

i.e., *all* non-zero residual directions. By construction, $X_{t,\mathrm{tail}}^\ell = 0$, so Lemma 3.1 forces $\|(W_t^\ell - W_k^\ell)\, x_k^\ell(k)\|_F = 0$ for every past task and every layer, hence zero forgetting.

Why this is unworkable in practice. The cost is twofold. First, the construction *abandons the low-rank premise* of LoRA: $r^\star = \mathrm{rank}(X_{t,\mathrm{res}}^\ell)$ is not a small number. In our experiments on LLaMA-2 7B ($d = 4096$), each task's residual activations

have effective rank in the hundreds at every adapted layer—one to two orders of magnitude above any practical LoRA budget—so each adapter alone would already exceed the parameter footprint that PEFT was meant to deliver. Second, even granting an arbitrarily large per-task budget, the orthogonality constraint exhausts the hidden space: the cumulative basis dimension $\sum_{i\le t} r_i$ is bounded by $d - \operatorname{rank}\big(\bigcup_{k<t} \operatorname{Col}(x_k^\ell(k))\big)$, which collapses to 0 once past activations span the full d-dimensional hidden space, after which no further adaptation is possible.

The relaxation (2) avoids this collapse: every direction is graded by its singular value, and only the top of the spectrum is paid for in rank.

D. Proof of Lemma 3.1 (Bounded layerwise interference)

Proof. Fix a layer (superscript ℓ omitted) and a past task k. Recall that the accumulated subspace through task k is $\mathcal{S}_{1:k} = \operatorname{span}\big(\bigcup_{i=1}^{k} \operatorname{Row}(B_i)\big)$, and by construction $\operatorname{Row}(B_k) \subseteq (\mathcal{S}_{1:k-1})^\perp$.

Step 1 (decomposition of $x_k(k)$). Decompose the time-k activation as

$$x_k(k) \;=\; \underbrace{\sum_{i=1}^{k-1} B_i^\top B_i\, x_k(k) \;+\; B_k^\top B_k\, X_{k,\text{res}}}_{\in\, \mathcal{S}_{1:k}} \;+\; X_{k,\text{tail}},$$

where $X_{k,\text{tail}} = X_{k,\text{res}} - B_k^\top B_k\, X_{k,\text{res}} = P_{1:k}^\perp\, x_k(k) \in \mathcal{S}_{1:k}^\perp$.

Step 2 (future adapters annihilate the $\mathcal{S}_{1:k}$ component). For any future task $i > k$, the orthogonality constraint of the construction gives $\operatorname{Row}(B_i) \perp \mathcal{S}_{1:i-1} \supseteq \mathcal{S}_{1:k}$, so B_i annihilates the $\mathcal{S}_{1:k}$ part and

$$\Delta W_i\, x_k(k) \;=\; A_i B_i\, x_k(k) \;=\; A_i B_i\, X_{k,\text{tail}}.$$

Step 3 (triangle inequality and submultiplicativity). Summing over $i = k{+}1, \dots, t$,

$$\|(W_t - W_k)\, x_k(k)\|_F \;=\; \Big\| \sum_{i=k+1}^{t} A_i B_i\, X_{k,\text{tail}} \Big\|_F \;\le\; \sum_{i=k+1}^{t} \|A_i\|_2\, \|B_i\, X_{k,\text{tail}}\|_F \;\le\; \Big(\sum_{i=k+1}^{t} \|A_i\|_2 \Big)\, \|X_{k,\text{tail}}\|_F,$$

where the last inequality uses $\|B_i\|_2 = 1$ (orthonormal rows). This proves (4).

Step 4 (optimality of the top-r SVD basis). Among all B_k with r_k orthonormal rows lying in $(\mathcal{S}_{1:k-1})^\perp$, $\|X_{k,\text{tail}}\|_F^2 = \|X_{k,\text{res}} - B_k^\top B_k X_{k,\text{res}}\|_F^2$ is exactly the error of projecting $X_{k,\text{res}}$ onto an r_k-dimensional subspace of $(\mathcal{S}_{1:k-1})^\perp$. Since $X_{k,\text{res}} = P_{1:k-1}^\perp X_k$ lies in $(\mathcal{S}_{1:k-1})^\perp$ already, this subspace constraint is inactive, and by the Eckart–Young–Mirsky theorem the error is minimized by taking B_k to be the top-r_k left singular vectors of $X_{k,\text{res}}$, giving $\|X_{k,\text{tail}}\|_F^2 = \sum_{j>r_k} \sigma_{k,j,\text{res}}^2$. □

E. Proof of the end-to-end drift expression (3)

We prove the composition of per-layer bounds into the end-to-end drift (3) in two steps: an attention-block drift lemma, and a transformer-block drift lemma. The statements are slightly more detailed than in the main text; the bound (3) follows by unrolling.

Notation. For the block at layer l, write the forward computation as $U^l \to Z^l = \mathrm{LN}_1^l(U^l) \to R^l = U^l + \mathrm{Att}^l(Z^l) \to Y^l = \mathrm{LN}_2^l(R^l) \to U^{l+1} = R^l + \mathrm{MLP}^l(Y^l)$. Under adapter state $\mathcal{A}_s$ processing task-k data, denote this by $U_s^l, Z_s^l, \dots$.

Lemma E.1 (Attention-block drift)**.** *Fix a layer and head dimension d_h. For $s \in \{k, t\}$ let $Q_s = W_{Q,s}\, Z_s$, $K_s = W_{K,s}\, Z_s$, $V_s = W_{V,s}\, Z_s$. Define*

$$\alpha_V := \big\|\mathrm{sm}(Q_k^\top K_k/\sqrt{d_h})\big\|_2, \quad \alpha_Q := L_{\mathrm{sm}} \|V_t\|_2 \tfrac{\|K_t\|_2}{\sqrt{d_h}}, \quad \alpha_K := L_{\mathrm{sm}} \|V_t\|_2 \tfrac{\|Q_k\|_2}{\sqrt{d_h}},$$

where L_{sm} is the row-wise softmax Lipschitz constant. Then $\|\mathrm{Att}_t(Z_t) - \mathrm{Att}_k(Z_k)\|_F \le B_{\mathrm{att}} + M_{\mathrm{att}} \|Z_t - Z_k\|_F$ with

$$B_{\mathrm{att}} := \sum_{\Pi \in \{Q,K,V\}} \alpha_\Pi \big\|(W_{\Pi,t} - W_{\Pi,k}) Z_k\big\|_F, \qquad M_{\mathrm{att}} := \sum_{\Pi \in \{Q,K,V\}} \alpha_\Pi \|W_{\Pi,t}\|_2.$$

Proof. Let $O_s = V_s\,\mathrm{sm}(Q_s^\top K_s/\sqrt{d_h})^\top$. Writing $O_t - O_k = V_t(A_t - A_k)^\top + (V_t - V_k)A_k^\top$ with $A_s := \mathrm{sm}(Q_s^\top K_s/\sqrt{d_h})$ and applying submultiplicativity yields $\|O_t - O_k\|_F \le \|V_t\|_2\|A_t - A_k\|_F + \|A_k\|_2\|V_t - V_k\|_F$. Lipschitzness of softmax combined with the triangle inequality on $Q_t^\top K_t/\sqrt{d_h} - Q_k^\top K_k/\sqrt{d_h}$ gives $\|A_t - A_k\|_F \le L_{\mathrm{sm}}(\|K_t\|_2\|Q_t - Q_k\|_F + \|Q_k\|_2\|K_t - K_k\|_F)/\sqrt{d_h}$. Combining, $\|O_t - O_k\|_F \le \alpha_V\|V_t - V_k\|_F + \alpha_Q\|Q_t - Q_k\|_F + \alpha_K\|K_t - K_k\|_F$. Finally, for each $\Pi \in \{Q, K, V\}$ decompose $\Pi_t - \Pi_k = (W_{\Pi,t} - W_{\Pi,k})Z_k + W_{\Pi,t}(Z_t - Z_k)$ and apply the triangle inequality; grouping yields $B_{\mathrm{att}} + M_{\mathrm{att}}\|Z_t - Z_k\|_F$. □

Lemma E.2 (Transformer-block drift)**.** *Assume* $\mathrm{LN}_1^l, \mathrm{LN}_2^l$ *are* $L_{\mathrm{LN},1}^l, L_{\mathrm{LN},2}^l$*-Lipschitz and* MLP^l *(frozen) is* L_{MLP}^l*-Lipschitz. Then*

$$\|U_t^{l+1} - U_k^{l+1}\|_F \;\le\; a_l\,\|U_t^l - U_k^l\|_F + b_l,$$

with $a_l := (1 + L_{\mathrm{MLP}}^l L_{\mathrm{LN},2}^l)(1 + M_{\mathrm{att}}^l L_{\mathrm{LN},1}^l)$ *and* $b_l := (1 + L_{\mathrm{MLP}}^l L_{\mathrm{LN},2}^l)\,B_{\mathrm{att}}^l$.

Proof. By Lipschitzness of LN_1^l and Lemma E.1, $\|\mathrm{Att}_t^l(Z_t^l) - \mathrm{Att}_k^l(Z_k^l)\|_F \le B_{\mathrm{att}}^l + M_{\mathrm{att}}^l L_{\mathrm{LN},1}^l\|U_t^l - U_k^l\|_F$. After the first residual connection, $\|R_t^l - R_k^l\|_F \le (1 + M_{\mathrm{att}}^l L_{\mathrm{LN},1}^l)\|U_t^l - U_k^l\|_F + B_{\mathrm{att}}^l$. Since MLP^l is L_{MLP}^l-Lipschitz and LN_2^l is $L_{\mathrm{LN},2}^l$-Lipschitz, $\|\mathrm{MLP}^l(Y_t^l) - \mathrm{MLP}^l(Y_k^l)\|_F \le L_{\mathrm{MLP}}^l L_{\mathrm{LN},2}^l\|R_t^l - R_k^l\|_F$. The second residual connection then gives the claim. □

End-to-end drift. Since task-k data enter the network identically at time t and time k, $U_t^0 = U_k^0 = X_k$, so $\|U_t^0 - U_k^0\|_F = 0$. Unrolling Lemma E.2 through L blocks yields

$$\|U_t^L - U_k^L\|_F \;\le\; \sum_{i=0}^{L-1} b_i \prod_{j=i+1}^{L-1} a_j,$$

which is exactly (3) once b_i, a_j are expanded. The derivation up to this point is purely kinematic: at block i, each summand of B_{att}^i is of the form $\alpha_\Pi^i\,\|(W_{\Pi,t} - W_{\Pi,k})Z_k\|_F$, which is exactly the generic per-layer drift controlled by the tolerance $\varepsilon_k^{\Pi,i}$ introduced in Section 2. Specializing to the residual-subspace construction of Section 3 and applying Lemma 3.1 to each attention projection $\Pi \in \{Q, K, V\}$ bounds each summand by the corresponding tail norm $\|X_{k,\mathrm{tail}}^{\Pi,i}\|_F$ up to training-dependent factors, so the threshold τ of (7) controls every term. □

F. Per-layer tail bound from the threshold rule

Remark F.1 (τ as a uniform per-layer tail cap). Under the threshold rule (7), every singular value uncaptured at layer ℓ satisfies $\sigma_{k,j,\mathrm{res}}^\ell < \tau$, so

$$\|X_{k,\mathrm{tail}}^\ell\|_F^2 \;=\; \sum_{j > r_{\tau,k}^\ell} (\sigma_{k,j,\mathrm{res}}^\ell)^2 \;\le\; \tau^2\big(\rho_k^\ell - r_{\tau,k}^\ell\big). \tag{8}$$

The bound holds at *every* adapted layer simultaneously: the single dual variable τ doubles as a uniform per-layer tail cap, and lowering τ shrinks every per-layer injection of (3) and hence the end-to-end forgetting bound. A fixed global rank r provides no analogous guarantee: two layers with very different residual spectra leave very different tail masses uncovered, so the per-layer drift is bounded only by the worst-case spectrum-rich layer.

G. CS slack and alternative-norm surrogates

The Cauchy–Schwarz step of (5) bounds $\sum_\ell C_k^\ell\,T_\ell(r_k^\ell)$ by an ℓ^2/ℓ^2 Hölder splitting and absorbs the unknown C_k^ℓ into the rank-independent constant $K_k = \sum_\ell (C_k^\ell)^2$. This collapses the per-layer importance asymmetry into a single layer-blind constant: any two layers with the same residual spectrum receive the same allocation, regardless of their Lipschitz factors or post-hoc adapter norms.

The Hölder family. CS is the $(p, q) = (2, 2)$ instance of Hölder's inequality. The $(p, q) = (\infty, 1)$ endpoint gives the alternative

$$\textstyle\sum_\ell C_k^\ell\,T_\ell \;\le\; \max_\ell C_k^\ell \cdot \sum_\ell T_\ell,$$

whose rank-independent constant is $\max_\ell C_k^\ell$ and whose surrogate is the sum of per-layer tail norms $\sum_\ell \|X_{k,\mathrm{tail}}^\ell\|_F$. This is tighter than CS when one C_k^ℓ dominates the others (worst-case ratio $\sqrt{L}$ in either direction). The resulting allocation

problem is separable across layers but *not* across per-layer directions (ℓ, j): the optimal rule allocates rank by the per-layer marginal $\partial T_\ell / \partial r$, not by individual singular values, so the single-threshold structure of (7) is lost.

Partial recovery via the threshold rule. Although CS suppresses C_k^ℓ in the surrogate, the threshold rule still gives a per-layer guarantee that does not depend on C_k^ℓ: by (8), $\|X_{k,\text{tail}}^\ell\|_F^2 \le \tau^2(\rho_k^\ell - r_{\tau,k}^\ell)$ at every layer simultaneously, so each per-layer injection of (3) is bounded uniformly. The surrogate's layer asymmetry is collapsed; the resulting rule's per-layer guarantee is not.

Truly layer-weighted alternatives. A more ambitious direction is to estimate C_k^ℓ rather than bound it away. Possibilities include: post-hoc correction of the threshold once $\sum_{i>k} \|A_i^\ell\|_2$ is observable; cheap proxies for the Lipschitz factors at allocation time (e.g., spectral norms of frozen weights); or curriculum-style task ordering that controls the post-hoc factor in expectation. We leave these to future work.

H. Heterogeneous per-rank costs

In Section 4 we work with the total-rank budget $\sum_\ell r_k^\ell \le R_k$, valid when every adapted layer incurs the same number of parameters per unit of rank (the W_Q, W_V projections of LLaMA-2 7B in our experiments). In general, each unit of rank at layer ℓ costs $w_\ell := d_{\text{in}}^\ell + d_{\text{out}}^\ell$ parameters (LoRA: $A \in \mathbb{R}^{d_{\text{out}} \times r}$, $B \in \mathbb{R}^{r \times d_{\text{in}}}$), so the budget constraint becomes $\sum_\ell w_\ell \, r_k^\ell \le W_k$.

The squared-tail surrogate of (6) remains separable across (ℓ, j), but the Lagrangian greedy solution now sorts directions by their reward-per-cost ratio $(\sigma_{k,j,\text{res}}^\ell)^2 / w_\ell$. With dual variable $\tau \ge 0$, the threshold rule (7) becomes layer-weighted:

$$r_k^\ell \;=\; \#\big\{\, j : \sigma_{k,j,\text{res}}^\ell \ge \tau \sqrt{w_\ell} \,\big\}. \tag{9}$$

When w_ℓ is constant, $\sqrt{w_\ell}$ absorbs into τ and we recover (7). Operationally, it suffices to scale each layer's singular values by $1/\sqrt{w_\ell}$ before the global threshold selection.

I. Reproducibility details

This appendix consolidates the training and evaluation settings used in Section 5. Table 1 gives a single-table summary; per-item justifications follow.

Table 1. Summary of training and evaluation settings for all experiments in Section 5.

Item	Value
Base model	LLaMA-2 7B (Touvron et al., 2023), `bfloat16`, frozen
Adapted projections	W_Q and W_V of every self-attention layer (64 total)
Optimiser	AdamW
Learning rate	10^{-4}
LR scheduler / warmup	None
Weight decay	0.01
Epochs per task	5
Batch size	2 (1 for MeetingBank)
Max sequence length	4,096 tokens (longer sequences skipped)
Gradient checkpointing	Enabled
Hardware	NVIDIA A100 (40 GB / 80 GB)
Seeds	$\{42, 123, 456, 789, 1024\}$
Benchmark	TRACE (Wang et al., 2023b), default task order
Tasks (T)	8 (C-STANCE, FOMC, MeetingBank, Py150, ScienceQA, NumGLUE-cm, NumGLUE-ds, 20Minuten)
Threshold sweep τ	$\{1.8, 2.0, 2.4, 2.8, 3.2, 3.6\}$
Fixed-rank sweep r	$\{4, 6, 8, 10, 14, 18\}$
Reported metrics	Average Performance (AP), Backward Transfer (BWT)
Statistics	Mean $\pm$ std over 5 seeds

Model and adapters. All experiments use LLaMA-2 7B (Touvron et al., 2023) in `bfloat16` with frozen pretrained weights. Adapters are inserted at the query (W_Q) and value (W_V) projections of every self-attention layer (64 adapted

projections across 32 transformer layers). For both InfLoRA and DyRA, only the output projection A_t^ℓ is trained; the input basis B_t^ℓ is computed from the residual-activation SVD of Section 4 and frozen thereafter.

Optimiser and schedule. We train A_t^ℓ for 5 epochs per task with AdamW, learning rate 10^{-4} and no learning-rate schedule (hence no warmup).

Batching and context length. The batch size is 2 for all tasks except MeetingBank, which uses batch size 1 due to longer transcripts. Sequences exceeding 4,096 tokens are skipped. Gradient checkpointing is enabled throughout.

Seeds. All experiments are repeated over 5 random seeds $\{42, 123, 456, 789, 1024\}$.

Statistical reporting. Unless stated otherwise, we report the empirical mean and one empirical standard deviation σ across the 5 seeds. The standard error of the mean is $\sigma/\sqrt{5} \approx 0.447\,\sigma$; the tables and figures use $\pm\sigma$ (std), the more conservative choice.

Hardware. All runs are performed on NVIDIA A100 GPUs (40 GB and 80 GB variants).

Evaluation metrics. After learning each task t, the model is evaluated on all tasks $1, \ldots, t$, producing the upper-triangular matrix $R \in \mathbb{R}^{T\times T}$ with $R_{t,k}$ the score on task k after training through task t. We report $\mathrm{AP} := \frac{1}{T}\sum_{k=1}^{T} R_{T,k}$ (mean end-of-sequence score) and $\mathrm{BWT} := \frac{1}{T-1}\sum_{k=1}^{T-1}(R_{T,k} - R_{k,k})$ (average performance change on past tasks; 0 means no forgetting, negative means degradation). Per-task metrics follow the TRACE defaults.

Hyperparameter sweeps. The threshold sweep is $\tau \in \{1.8, 2.0, 2.4, 2.8, 3.2, 3.6\}$; the fixed-rank baseline sweep is $r \in \{4, 6, 8, 10, 14, 18\}$. These are the configurations shown in Figure 1 and Table 2. The continual-LoRA baseline uses the same r sweep and the same optimiser / schedule / seeds as the orthogonal-subspace methods; the only difference is that both A_t and B_t are trained, with no residual-subspace constraint.

Backward transfer at matched capacity. Table 2 compares BWT for $\tau = 3.2$ ($\bar{r} \approx 5.5$) vs. fixed $r = 6$, and $\tau = 2.8$ ($\bar{r} \approx 7.4$) vs. fixed $r = 8$. Thresholding matches or slightly improves BWT at matched capacity, and never hurts.

Table 2. BWT at matched capacity on TRACE (LLaMA-2 7B, 5 seeds). Thresholding matches or improves BWT while using equal or fewer parameters.

Method	Avg rank	BWT
Fixed $r = 6$	6.00	-0.0090 ± 0.0267
DyRA ($\tau = 3.2$)	5.49 ± 0.08	-0.0065 ± 0.0247
Fixed $r = 8$	8.00	-0.0159 ± 0.0153
DyRA ($\tau = 2.8$)	7.37 ± 0.05	-0.0044 ± 0.0155

Full AP sweep with uncertainty. Table 3 reports the mean AP and one empirical standard deviation across seeds for all three methods (threshold DyRA, fixed-rank InfLoRA, and continual LoRA without projection) over the entire rank/threshold sweep. These are the numbers underlying Figure 1.

Table 3. Average Performance (AP) across TRACE tasks for the full sweep, reported as mean $\pm$ one empirical standard deviation across seeds. Orthogonal-subspace methods (DyRA, Fixed InfLoRA) dominate continual LoRA by ~ 15 percentage points at every rank. DyRA is at least on par at every matched-capacity point, with std-exceeding gains at low rank ($\bar{r} \leq 6$).

Method	**Config**	**Avg rank $\bar{r}$**	**AP (%)**
DyRA (ours)	$\tau = 3.6$	4.22 ± 0.04	38.73 ± 1.17
	$\tau = 3.2$	5.49 ± 0.08	41.17 ± 2.45
	$\tau = 2.8$	7.37 ± 0.05	42.10 ± 1.33
	$\tau = 2.4$	10.05 ± 0.08	42.99 ± 1.20
	$\tau = 2.0$	14.01 ± 0.24	44.12 ± 1.19
	$\tau = 1.8$	17.23 ± 0.30	45.00 ± 0.99
Fixed InfLoRA	$r = 4$	4.00	34.07 ± 1.77
	$r = 6$	6.00	39.95 ± 2.19
	$r = 8$	8.00	39.57 ± 4.45
	$r = 10$	10.00	40.40 ± 3.33
	$r = 14$	14.00	43.13 ± 1.73
	$r = 18$	18.00	44.21 ± 2.39
Continual LoRA (no proj.)	$r = 4$	4.00	25.84 ± 3.21
	$r = 6$	6.00	26.14 ± 2.37
	$r = 8$	8.00	27.03 ± 3.47
	$r = 10$	10.00	26.91 ± 3.72
	$r = 14$	14.00	27.32 ± 3.28
	$r = 18$	18.00	28.26 ± 4.39