What Did My Adapter Break? A Generation-Level Evaluation of Adaptation and Retention in PEFT-Adapted LLMs

Guy Bilitski[1,2], Kfir Bar[1], Shai Fine[2]
[1]Efi Arazi School of Computer Science, Reichman University, Israel
[2]Data Science Institute, Reichman University, Israel
guy.bilitski@post.runi.ac.il, kfir.bar@runi.ac.il, shai.fine@runi.ac.il

Abstract

Parameter-efficient fine-tuning enables instruction-tuned large language models to adapt to new tasks and domains, but it can also destabilize outputs the base model previously generated reliably. Existing analyses commonly measure such degradation through aggregate benchmark deltas, which reveal overall performance changes but not which outputs fail or how severely. We introduce a dual-axis, generation-level evaluation framework that measures adaptation and retention through repeated generations. Our Negative Shift measure identifies prompts whose correct-generation rate collapses after adaptation. Using this framework, we compare LoRA, DoRA, VeRA, and our proposed UIOrthoLoRA across learning-rate and capacity sweeps on Llama-3.2-3B-Instruct and Gemma-3-12B-IT. The retention–adaptation trade-off is highly sensitive to learning rate: LoRA and DoRA exhibit abrupt degradation under aggressive optimization, whereas structurally constrained parameterizations degrade more gradually. At comparable target performance, moving from small to medium adapter capacity often improves retention, although gains are not monotonic. UIOrthoLoRA, a spectrum-aware adapter that restricts directional adaptation to the low-energy spectral tail, improves the observed trade-off relative to the evaluated baselines under constrained parameter budgets.

1 Introduction

Instruction-tuned large language models (LLMs) are general-purpose natural language generation systems. When such a model must be adapted to specialized knowledge or a new application, parameter-efficient fine-tuning (PEFT), particularly Low-Rank Adaptation (LoRA; Hu et al., 2022), provides a practical alternative to updating the pre-trained parameters.

Successful adaptation, however, does not guarantee that the resulting generator remains reliable. An adapter may learn to produce the desired outputs for its target examples while changing answers that the base model previously generated correctly and consistently. These failures can be difficult to detect because the adapted model often remains fluent, responsive, and apparently capable. The degradation may therefore manifest not as a general loss of linguistic quality, but as a change in *which outputs the model can be trusted to generate*.

Studies of forgetting in adapted LLMs often quantify this problem through aggregate changes on external benchmarks. Such evaluations are valuable for measuring broad capability degradation, but they provide only a partial account of how adaptation changes generation behavior. A benchmark-level decrease indicates that overall performance has changed, but it does not identify the prompts for which previously reliable outputs failed. It may also obscure the distinction between small changes distributed across many examples and severe collapses concentrated on a smaller set of prompts. From an NLG perspective, this distinction is important: a model may retain much of its average performance while becoming unreliable on particular inputs that it previously handled consistently.

Inspired by Pletenev et al. (2025), we build on this sampling-based view of model knowledge and extend it in three directions. First, we formulate adaptation and retention as two explicit evaluation axes and use them to characterize the full retention–adaptation trade-off across adapter configurations. Second, building on the notion of negative knowledge transitions, we define a strict **Negative Shift** measure that isolates severe collapses in generation reliability. Third, we broaden the analysis beyond a single LoRA parameterization by comparing adapters with distinct structural mechanisms across learning-rate and capacity sweeps, includ-

Proceedings of the 19th International Natural Language Generation Conference: Long Papers, pages 498–514
October 17-21, 2026 ©2026 Association for Computational Linguistics

ing our proposed adapter. This allows us to study not only whether adaptation damages previously reliable generations, but also how adapter structure and optimization determine the trajectory of that degradation.

Our **dual-axis evaluation framework** measures both target adaptation and retention of previously reliable generations. Using repeated sampling, we construct model-specific sets of questions that the base model answers consistently incorrectly or consistently correctly. We evaluate target-answer acquisition on the former and measure severe prompt-level degradation on the latter through Negative Shift. We complement this intrinsic measure with MMLU and nine BIG-bench multiple-choice tasks to assess broader knowledge and reasoning retention.

We introduce **UIOrthoLoRA**, a spectrum-aware adapter designed to preserve dominant pre-trained singular directions while concentrating its most flexible adaptation in the low-energy spectral tail. This structure aims to support target adaptation while limiting disruption to previously reliable model behavior.

Using our dual-axis framework, we compare UIOrthoLoRA with LoRA, DoRA, and VeRA across learning-rate and capacity sweeps on Llama-3.2-3B-Instruct and Gemma-3-12B-IT, using model-specific subsets derived from TriviaQA and HotpotQA. Our results show that retention is strongly affected by both optimization and adapter structure. LoRA and DoRA exhibit abrupt degradation under aggressive learning rates, whereas VeRA and UIOrthoLoRA degrade more gradually. Among configurations achieving comparable target performance, higher-capacity adapters generally preserve more of the base model's prior generation behavior. Across the evaluated models and datasets, UIOrthoLoRA improves the observed retention–adaptation trade-off relative to the evaluated baselines, with the clearest advantage under constrained parameter budgets.

Our main contributions are:

- **A generation-level evaluation framework** that jointly measures target adaptation and retention of previously reliable generations using model-specific error and known sets.
- **A prompt-level Negative Shift measure** that identifies severe collapses in correct-generation reliability after adaptation.
- **A systematic analysis of PEFT adaptation dynamics** across adapter mechanisms, learning rates, and capacity tiers.
- **UIOrthoLoRA**, a new spectrum-aware adapter that concentrates adaptation in the low-energy spectral tail and improves the observed retention–adaptation trade-off relative to the evaluated baselines, particularly at small capacity.
- **Reusable evaluation artifacts**, including model-specific splits, generation scores, and learning-rate and capacity sweep results.

Research questions. We organize our analysis around three questions: **RQ1**, how learning rate and adapter capacity affect the retention–adaptation trade-off; **RQ2**, how different adapter parameterizations shape the severity and trajectory of generation degradation; and **RQ3**, whether spectrum-aware structural constraints improve retention at comparable adaptation performance.

2 Related Work

Forgetting in Adapted Language Models. Freezing pre-trained parameters does not guarantee preservation of the base model's behavior, because the adapter still changes the function computed by the model. Prior work documents degradation of pre-trained capabilities after downstream adaptation (Luo et al., 2023; Kotha et al., 2024). In factual adaptation, Pletenev et al. (2025) estimate model-specific knowledge through repeated generations and track transitions between Unknown, Maybe-Known, and HighlyKnown states after LoRA adaptation. Their results show that learning new facts can disrupt previously reliable answers and reduce external benchmark performance. Biderman et al. (2024) further show that LoRA generally forgets less than full fine-tuning, but may also learn less when the target update is difficult to express at low rank. These findings motivate evaluating adaptation and retention jointly.

PEFT Mechanisms. LoRA (Hu et al., 2022) represents the update through two trainable low-rank factors. DoRA (Liu et al., 2024) separates weight magnitude from direction, while VeRA (Kopiczko et al., 2024) learns scaling vectors over frozen random projections. We compare these methods because they represent distinct update parameterizations: learned low-rank directions, magnitude–direction decomposition, and scaling over fixed

projections. Our goal is not to rank all PEFT variants, but to examine how adapter structure affects the retention–adaptation trade-off.

Retention-Aware and Spectral Adaptation. Several methods constrain adaptation through subspace selection or spectral structure. MiLoRA (Wang et al., 2025) initializes updates from minor singular components, while CorDA (Yang et al., 2024), SC-LoRA (Luo et al., 2025), LoRA-Null (Tang et al., 2026), and CLoRA (Lu et al., 2025) use activation- or null-space-based constraints to reduce interference. O-LoRA (Wang et al., 2023) assigns successive tasks to orthogonal low-rank subspaces. Related spectral methods include PiSSA (Meng et al., 2024), which adapts principal singular components, LoRA-XS (Bałazy et al., 2024), which freezes singular bases and learns a compact transformation, and LASER (Sharma et al., 2024), which removes lower-order components.

UIOrthoLoRA aims to observe the entire training dynamic through an SVD spectrum lens. It differs from existing methods by partitioning the pretrained spectrum into regions with distinct adaptation regimes. It preserves dominant singular directions, permits magnitude adaptation in intermediate components, and restricts its most flexible directional changes to rotations within the low-energy tail. These rotations remain within the selected tail subspaces before channel-wise feature scaling. The architecture also incorporates unit-invariant feature scalers following Uhlmann (2018).

3 Methodology

This section first presents our generation-level framework for measuring target adaptation and retention of previously reliable outputs. We then introduce UIOrthoLoRA, the spectrum-aware adapter evaluated under this framework.

3.1 Generation-Level Evaluation of Adaptation and Retention

We evaluate an adapted model along two complementary axes. The *adaptation axis* measures whether the model learns to generate answers that the base model consistently failed to produce. The *retention axis* measures whether answers that were reliable before adaptation remain reliable afterward.

Repeated-Generation Score. Let x denote a question and $\mathcal{A}(x)$ its set of reference answers. TriviaQA distributes several accepted surface forms (aliases) of each answer, and $\mathcal{A}(x)$ contains all of them; HotpotQA provides a single gold answer string. Given a model M, we independently sample $N = 10$ responses, denoted by $\{\hat{y}_M^{(i)}(x)\}_{i=1}^N$. Each response is matched against $\mathcal{A}(x)$ using the answer-normalization and matching procedure specified in Appendix B. We define the empirical correct-generation rate as

$$S_M(x) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[\hat{y}_M^{(i)}(x) \text{ matches } \mathcal{A}(x)\right]. \quad (1)$$

Thus, $S_M(x) \in \{0, 0.1, \ldots, 1.0\}$ estimates how reliably model M generates a correct answer under the fixed prompting and decoding protocol. We write $S_{\text{pre}}(x)$ for the frozen base-model score and $S_{\text{post}}(x)$ for the post-adaptation score.

Adaptation Axis. For each base model and source dataset, we construct a model-specific *error set*:

$$\mathcal{D}_{\text{err}} = \{x : S_{\text{pre}}(x) = 0\}. \quad (2)$$

This set contains questions for which the base model produces no correct answer across the ten sampled generations. We derive $\mathcal{D}_{\text{err}}$ from the training splits of TriviaQA (Joshi et al., 2017) and HotpotQA (Yang et al., 2018), and train adapters exclusively on the corresponding error set.

We measure target adaptation using the mean post-adaptation correct-generation rate:

$$\text{Adaptation} = \frac{1}{|\mathcal{D}_{\text{err}}|} \sum_{x \in \mathcal{D}_{\text{err}}} S_{\text{post}}(x). \quad (3)$$

Because every example in $\mathcal{D}_{\text{err}}$ has a pre-adaptation score of zero, and because adapters are trained on $\mathcal{D}_{\text{err}}$ itself, this quantity measures the model's acquisition of the target answers on the training examples; it does not measure generalization to held-out questions.

Retention Axis. We evaluate intrinsic retention on the model-specific *known set*:

$$\mathcal{D}_{\text{known}} = \{x : S_{\text{pre}}(x) = 1\}. \quad (4)$$

These are prompts for which the base model generates a correct answer in all ten samples. Restricting retention evaluation to this set distinguishes disruption of previously reliable outputs from ordinary uncertainty on prompts that the base model did not answer consistently. We use the strictest criterion, $S_{\text{pre}}(x) = 1$, because it keeps attribution

clean: a later collapse on these prompts reflects a change in behavior rather than pre-existing uncertainty. Relaxing the criterion (for example, admitting prompts with $S_{\text{pre}}(x) \geq 0.9$) would enlarge $\mathcal{D}_{\text{known}}$ at the cost of including prompts whose degradation is harder to separate from sampling variability; Appendix G examines the analogous choice on the collapse-threshold side.

Negative Shift. Aggregate accuracy can conceal severe changes affecting individual prompts. For $x \in \mathcal{D}_{\text{known}}$, we define a prompt-level Negative Shift event as

$$\mathbb{I}_{\text{NS}}(x) = \mathbb{I}[S_{\text{pre}}(x) - S_{\text{post}}(x) \geq \tau]\,. \quad (5)$$

We set $\tau = 0.8$. Because every prompt in $\mathcal{D}_{\text{known}}$ has $S_{\text{pre}}(x) = 1$, a Negative Shift occurs when $S_{\text{post}}(x) \leq 0.2$. It therefore captures a transition from consistently correct to predominantly incorrect generation while excluding smaller changes that may arise from sampling variability. Appendix G shows that the principal comparative trends remain similar under an alternative threshold.

We aggregate these events into a severity-weighted total, where each event is weighted by the size of its collapse in the correct-generation rate:

$$\text{NS} = \sum_{x \in \mathcal{D}_{\text{known}}} \mathbb{I}_{\text{NS}}(x)\,[S_{\text{pre}}(x) - S_{\text{post}}(x)]\,. \quad (6)$$

Because every counted event satisfies $S_{\text{pre}}(x) - S_{\text{post}}(x) \geq \tau = 0.8$, this total closely tracks the number of severely degraded prompts, and we refer to it simply as the Negative Shift of a configuration; reported values are given as whole numbers. Lower values indicate stronger retention of previously reliable outputs. Because $|\mathcal{D}_{\text{known}}|$ differs across models and datasets (Table 1), we also report each value as a percentage of $|\mathcal{D}_{\text{known}}|$, which places all settings on a common scale.

External Retention. To assess broader capability degradation, we additionally evaluate zero-shot accuracy on MMLU (Hendrycks et al., 2021) and **BIG-bench-9**, comprising nine BIG-bench multiple-choice tasks (Srivastava et al., 2023). The included tasks are listed in Appendix A.

When reporting a single external-retention score, we use their unweighted mean:

$$R_{\text{ext}} = \frac{\text{MMLU} + \text{BIG-bench-9}}{2}. \quad (7)$$

Negative Shift localizes severe changes to individual prompts, while MMLU and BIG-bench-9 measure broader degradation in knowledge and reasoning.

Retention–Adaptation Trade-Off. For each trained configuration, we plot adaptation against Negative Shift and against R_{ext}. Higher adaptation, lower Negative Shift, and higher external retention are preferred. These plots characterize how generation behavior changes across adapter mechanisms, learning rates, and capacity tiers.

3.2 UIOrthoLoRA

Standard PEFT adapters, including LoRA, DoRA, and VeRA, parameterize their updates without explicitly accounting for the singular structure of the pre-trained weight matrix. Related retention-aware and spectral methods often use subspace information to initialize or constrain a low-rank update. UIOrthoLoRA instead uses the pre-trained singular bases to define the core coordinate system of the adapter throughout training.

The central idea is to assign progressively greater adaptation flexibility to lower-energy spectral components. Within the core spectral parameterization, dominant singular directions remain fixed, intermediate directions permit coefficient adjustment, and the most flexible directional changes are restricted to orthogonal transformations within the low-energy spectral tail. This design aims to preserve the high-energy structure of the pre-trained matrix while using less emphasized directions for task-specific adaptation. Figure 1 summarizes the architecture.

For a frozen pre-trained weight matrix $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, we compute the thin SVD

$$W_0 = U\,\text{diag}(\sigma)V^\top, \quad (8)$$

where $U \in \mathbb{R}^{d_{\text{out}} \times r}$ and $V \in \mathbb{R}^{d_{\text{in}} \times r}$ have orthonormal columns, and $\sigma \in \mathbb{R}^r$, with $r = \min(d_{\text{out}}, d_{\text{in}})$ and $\sigma_1 \geq \cdots \geq \sigma_r \geq 0$. We partition the singular bases into dominant, intermediate, and tail regions:

$$\begin{aligned} U &= [U_{\text{maj}}\ U_{\text{mid}}\ U_{\text{tail}}]\,, \\ V &= [V_{\text{maj}}\ V_{\text{mid}}\ V_{\text{tail}}]\,. \end{aligned} \quad (9)$$

Two hyperparameters control the partition: the blocks contain $r - k_{\text{val}}$, $k_{\text{val}} - k_{\text{vec}}$, and k_{vec} columns, respectively, where $0 \leq k_{\text{vec}} \leq k_{\text{val}} \leq r$. The trailing k_{val} components have trainable adapter coefficients, while the trailing k_{vec} components

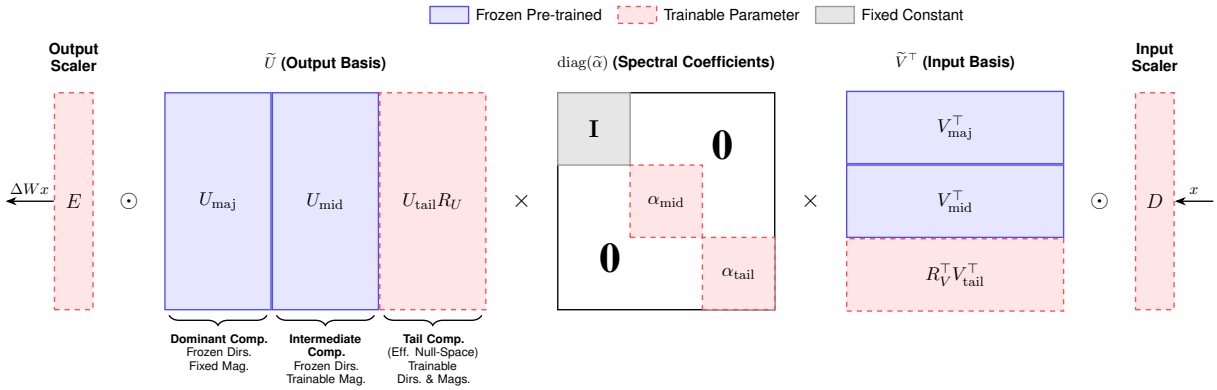


Figure 1: Architectural overview of **UIOrthoLoRA**. The pre-trained weight matrix is decomposed into its singular components (U, $\mathrm{diag}(\sigma)$, $V^\top$) and partitioned into the dominant, intermediate, and tail regions of Section 3.2. The dominant components are frozen, and the identity block $\mathbf{I}$ fixes their coefficients to one. Intermediate components have trainable spectral coefficients (α_{mid}). The tail, treated as an effective null space, has trainable coefficients (α_{tail}) and trainable rotations (R_U, R_V). Global scaling vectors (D, E) modulate the unit-invariance.

additionally permit directional adaptation. Here, α_{mid} and α_{tail} denote free adapter coefficients, distinct from the corresponding singular values of W_0.

UIOrthoLoRA applies a different adaptation regime to each region:

- **Dominant region.** The singular bases remain frozen and their coefficients in the core spectral parameterization are fixed to one. This region therefore introduces neither new spectral directions nor region-specific trainable spectral coefficients.
- **Intermediate region.** The singular bases remain frozen, while their spectral coefficients $\alpha_{\mathrm{mid}} \in \mathbb{R}^{k_{\mathrm{val}}-k_{\mathrm{vec}}}$ are trainable. The core adapter may therefore vary its contribution along these existing spectral directions while keeping their bases fixed.
- **Tail region.** The low-energy tail is treated as an effective null-space region. Its spectral coefficients $\alpha_{\mathrm{tail}} \in \mathbb{R}^{k_{\mathrm{vec}}}$ are trainable, and its bases are adapted within their existing subspaces through trainable orthogonal transformations $R_U, R_V \in \mathrm{O}(k_{\mathrm{vec}})$.

The resulting adapter bases and coefficients are

$$\begin{aligned} \widetilde{U} &= [U_{\mathrm{maj}}\ U_{\mathrm{mid}}\ U_{\mathrm{tail}} R_U]\,, \\ \widetilde{V} &= [V_{\mathrm{maj}}\ V_{\mathrm{mid}}\ V_{\mathrm{tail}} R_V]\,, \\ \widetilde{\alpha} &= \begin{bmatrix} \mathbf{1}_{r-k_{\mathrm{val}}} \\ \alpha_{\mathrm{mid}} \\ \alpha_{\mathrm{tail}} \end{bmatrix}. \end{aligned} \tag{10}$$

With trainable input and output feature scalers $D \in \mathbb{R}^{d_{\mathrm{in}}}$ and $E \in \mathbb{R}^{d_{\mathrm{out}}}$, the adapter update is

$$\Delta W = \mathrm{diag}(E)\widetilde{U}\,\mathrm{diag}(\widetilde{\alpha})\widetilde{V}^\top\,\mathrm{diag}(D), \tag{11}$$

and the adapted layer computes

$$h = (W_0 + \Delta W)x. \tag{12}$$

The adapted tail bases remain orthogonal to the frozen dominant and intermediate bases. UIOrthoLoRA preserves the pre-trained spectral coordinate system in the higher-energy component, while concentrating its most flexible directional adaptation in the low-energy tail. Relative to the original singular bases, any cross-region mixing is introduced only through the outer feature scalers D and E.

Details of the one-time SVD, parameter initialization, orthogonal parameterization, sequential forward computation, and trainable-parameter accounting are provided in Appendix D.

4 Experiments and Results

We use the generation-level framework of Section 3.1 to study how PEFT changes previously reliable outputs, and we organize the analysis around our three research questions: the effects of optimization and capacity (RQ1), the differences among adapter mechanisms (RQ2), and whether spectrum-aware structural constraints mitigate the observed degradation (RQ3).

4.1 Experimental Setup

We compare UIOrthoLoRA with LoRA (Hu et al., 2022), DoRA (Liu et al., 2024), and VeRA

Model	Dataset	$\lvert\mathcal{D}_{\text{err}}\rvert$	$\lvert\mathcal{D}_{\text{known}}\rvert$
Llama-3.2-3B	TriviaQA	32,799	43,724
	HotpotQA	11,323	4,612
Gemma-3-12B	TriviaQA	20,469	28,147
	HotpotQA	10,938	4,723

Table 1: Model-specific error and known set sizes.

(Kopiczko et al., 2024). These baselines represent learned low-rank factors, magnitude–direction decomposition, and learned scaling over frozen random projections, respectively.

Experiments use Llama-3.2-3B-Instruct (AI @ Meta, 2024) and Gemma-3-12B-IT (Gemma Team, 2025). Each model is adapted on model-specific error sets derived from the training splits of TriviaQA (Joshi et al., 2017) and HotpotQA (Yang et al., 2018). Throughout the results, adaptation accuracy is therefore measured on the same examples used for training: it reflects target-answer acquisition rather than generalization to held-out questions (see Limitations). All adapters within a model–dataset setting use the same error and known sets. Table 1 reports their sizes.

Across the experimental grid, learning rates span 10^{-6} to 10^{-2}, with the exact values varying by model–dataset setting as detailed in Appendix C. We group configurations by trainable parameter count p into small ($p \leq 10$M), medium (10M$< p \leq 100$M), and large ($p > 100$M) tiers. VeRA is represented only in the small-capacity tier in our evaluated grid.

All adapters target the same attention and feed-forward projection modules. Complete training and inference settings are provided in Appendices A and C. We evaluate each configuration using adaptation accuracy on $\mathcal{D}_{\text{err}}$, Negative Shift (Equation 6) on $\mathcal{D}_{\text{known}}$, and zero-shot accuracy on MMLU and BIG-bench-9. Table 2 summarizes the main results.

4.2 RQ1: Effects of Optimization and Capacity

4.2.1 Learning-Rate Sensitivity

Figure 2 shows the complete learning-rate sweep for Gemma-3-12B-IT adapted on HotpotQA. Corresponding results for the other model–dataset settings are provided in Appendix E.

Learning rate strongly affects both adaptation and retention. Increasing the learning rate initially improves target-answer acquisition, but aggressive optimization can sharply increase Negative Shift and reduce external benchmark accuracy. High target accuracy alone is therefore insufficient to identify a reliable adapted model.

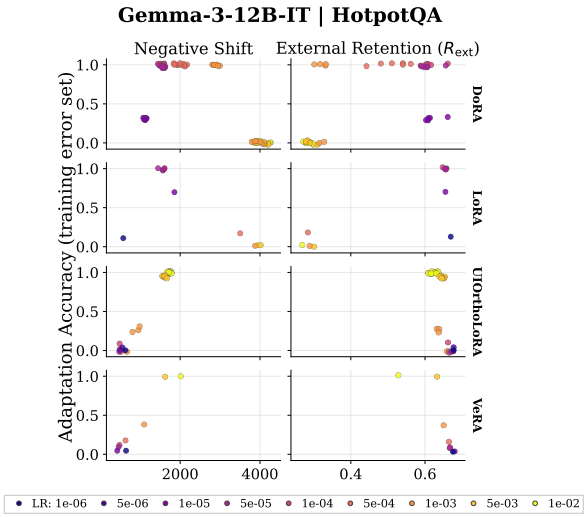


Figure 2: Learning-rate sweep for Gemma-3-12B-IT adapted on HotpotQA. Each row is one adapter method and each point is one trained configuration, with color encoding the learning rate; the vertical axis is adaptation accuracy on the training error set. The left column plots each configuration against its Negative Shift (lower is better) and the right column against external retention R_{ext} (higher is better), so reliable configurations appear in the upper left of the left panels and the upper right of the right panels.

The shape of this degradation differs across adapter mechanisms. LoRA and DoRA exhibit an abrupt transition: beyond a relatively narrow stable region, their Negative Shift rises rapidly and their external scores decline, with MMLU approaching 0.20 in the most degraded configurations. VeRA and UIOrthoLoRA follow a more gradual trajectory, with retention declining more smoothly as the strength of adaptation increases.

These results answer the optimization component of RQ1: the retention–adaptation trade-off is highly sensitive to learning rate, and the most aggressive configuration is not a reliable one.

4.2.2 Adapter Capacity

A common heuristic in PEFT practice is minimal necessary capacity: one should use the smallest adapter that fits the target task. This view is grounded in the intrinsic-dimensionality perspective of Aghajanyan et al. (2021), which shows that downstream optimization occurs in a low-dimensional subspace, and it is operationalized by methods such as AdaLoRA (Zhang et al., 2023), which prune parameter budgets during training.

Method	TriviaQA				HotpotQA			
	Adaptation	Retention			Adaptation	Retention		
	Acc ↑	Neg. Shift ↓	MMLU ↑	BIG-bench-9 ↑	Acc ↑	Neg. Shift ↓	MMLU ↑	BIG-bench-9 ↑
Gemma-3-12B-IT								
Base Model	–	*0*	*0.715*	*0.638*	–	*0*	*0.715*	*0.638*
Small								
VeRA	0.976	**8,463 (30.1%)**	0.691	0.607	**0.993**	1,624 (34.4%)	0.661	0.607
UIOrthoLoRA	0.982	8,523 (30.3%)	**0.696**	0.582	0.985	**1,548 (32.8%)**	0.678	**0.629**
LoRA	0.996	8,920 (31.7%)	0.691	0.611	0.990	1,589 (33.6%)	0.690	0.618
DoRA	**0.997**	8,825 (31.4%)	–	**0.625**	0.989	1,571 (33.3%)	**0.698**	0.624
Medium								
UIOrthoLoRA	0.982	8,430 (29.9%)	**0.704**	0.602	0.988	1,492 (31.6%)	0.691	0.597
LoRA	**0.997**	**7,825 (27.8%)**	0.703	**0.641**	**0.994**	**1,488 (31.5%)**	**0.699**	**0.615**
DoRA	**0.997**	8,114 (28.8%)	0.703	0.623	0.992	1,510 (32.0%)	0.696	0.611
Large								
UIOrthoLoRA	0.975	8,194 (29.1%)	0.702	0.604	0.988	1,523 (32.2%)	0.690	0.617
LoRA	**0.997**	7,639 (27.1%)	**0.705**	**0.637**	**0.995**	**1,497 (31.7%)**	**0.705**	**0.633**
DoRA	0.978	**7,604 (27.0%)**	**0.705**	0.627	0.993	1,514 (32.1%)	0.704	0.624
Llama-3.2-3B-Instruct								
Base Model	–	*0*	*0.622*	*0.458*	–	*0*	*0.622*	*0.458*
Small								
VeRA	0.925	13,805 (31.6%)	0.420	0.415	0.904	1,350 (29.3%)	0.250	0.353
UIOrthoLoRA	0.935	**10,088 (23.1%)**	**0.549**	**0.428**	**0.958**	**931 (20.2%)**	**0.530**	**0.419**
LoRA	0.938	13,597 (31.1%)	0.342	0.397	0.942	1,066 (23.1%)	0.496	0.404
DoRA	**0.952**	13,910 (31.8%)	0.422	0.388	0.954	1,064 (23.1%)	0.512	0.417
Medium								
UIOrthoLoRA	0.945	10,234 (23.4%)	0.544	0.440	0.960	**906 (19.6%)**	0.522	0.455
LoRA	0.961	8,967 (20.5%)	**0.601**	0.466	0.993	972 (21.1%)	0.548	0.459
DoRA	**0.963**	**8,908 (20.4%)**	**0.601**	**0.468**	**0.994**	982 (21.3%)	**0.561**	**0.461**
Large								
UIOrthoLoRA	0.943	10,108 (23.1%)	0.559	0.416	0.960	**870 (18.9%)**	0.549	0.449
LoRA	0.966	8,703 (19.9%)	0.601	0.456	0.994	892 (19.3%)	**0.581**	**0.462**
DoRA	**0.968**	**8,556 (19.6%)**	**0.603**	**0.467**	**0.996**	926 (20.1%)	0.570	0.455

Table 2: Balancing Adaptation and Retention (UIOrthoLoRA is our adapter). **Adaptation** is target accuracy (↑) on the training error set $\mathcal{D}_{\text{err}}$, the same examples used for training, so it reflects target-answer acquisition rather than held-out generalization; **Retention** is Negative Shift (Equation 6, ↓) on the known set $\mathcal{D}_{\text{known}}$ ($\tau = 0.8$; in parentheses, its value relative to $|\mathcal{D}_{\text{known}}|$, with split sizes in Table 1) together with zero-shot **MMLU** and **BIG-bench-9** accuracy (↑). Rows report representative configurations, so the capacity figures, which require adaptation ≥ 0.98, may show different runs for the same method and tier; bold marks the best value per column within each capacity tier, and a dash marks an evaluation that is unavailable or not applicable.

Under this view, a tighter bottleneck restricts deviation from the pre-trained solution and should therefore also protect prior behavior. Our matched-performance comparison tests this expectation directly.

To examine capacity while controlling for target performance, Figure 3 compares Gemma-3-12B-IT HotpotQA configurations achieving adaptation accuracy of at least 0.98. Corresponding comparisons for the other model–dataset settings are provided in Appendix F.

The clearest improvement occurs when moving from the small to the medium tier. LoRA's Negative Shift decreases from 1,589 to 1,468, while its external retention increases from 0.654 to 0.664. DoRA similarly decreases from 1,571 to 1,461, and UIOrthoLoRA decreases from 1,548 to 1,492 while its external retention rises from 0.653 to 0.675.

The large configurations remain competitive, but improvement is not monotonic for every method and metric. For example, UIOrthoLoRA achieves its highest external retention in the medium tier rather than the large tier. The results do not imply that the largest possible adapter is always preferable. Instead, they show that an overly narrow update can require greater interference to reach the same target accuracy, and that additional capacity can improve retention in the evaluated matched-performance comparisons.

Together, these findings answer the capacity component of RQ1: among configurations achiev-

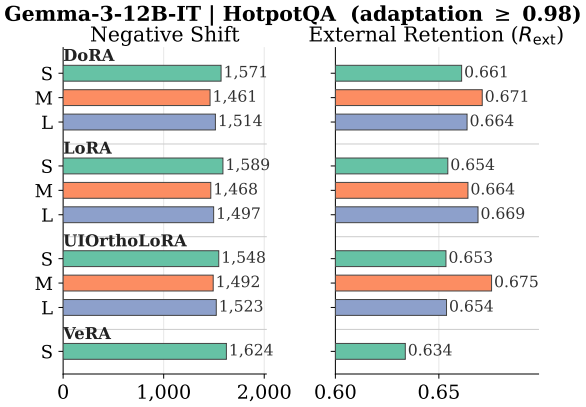


Figure 3: Capacity comparison on Gemma-3-12B-IT adapted on HotpotQA, grouped by adapter method and capacity tier (S, M, L); VeRA appears only in the small tier. Only configurations achieving adaptation accuracy ≥ 0.98 on the training error set and having complete external evaluations are included; for each method and tier, the left panel reports the lowest Negative Shift and the right panel the highest external retention R_{ext} among qualifying configurations, so the two panels may reflect different runs. Lower Negative Shift and higher R_{ext} are better.

ing similar adaptation performance, the smallest adapter does not necessarily better preserve prior behavior.

4.3 RQ2: Degradation Across Adapter Mechanisms

The baseline adapters also differ in how the retention loss develops across the optimization sweep. LoRA and DoRA learn update directions directly and exhibit the sharpest degradation at aggressive learning rates. VeRA restricts learning to scaling over frozen random projections and follows a more gradual trajectory, although it does not eliminate forgetting and is represented only in the small-capacity tier of our grid.

A plausible explanation for this divergence lies in the parameterization of the update. LoRA and DoRA learn their projection directions from scratch, so at aggressive learning rates, the learned directions can move the adapted function far from the pre-trained weight distribution. VeRA and UIOrthoLoRA instead operate over fixed vector spaces: VeRA scales frozen, randomly initialized projections, while UIOrthoLoRA adapts magnitudes over the frozen pre-trained spectral bases and confines its directional changes to rotations within the low-energy tail.

The intrinsic and external measures change together: configurations with large increases in Negative Shift also tend to show substantial declines on MMLU and BIG-bench-9. This agreement indicates that the prompt-level failures captured by Negative Shift accompany broader degradation in knowledge and reasoning.

These observations answer RQ2: existing adapter mechanisms do not provide uniform protection against forgetting, and the parameterization of the update affects the severity and trajectory of degradation. The contrast between learned-direction and structurally constrained adapters is consistent with the hypothesis that limiting directional flexibility can reduce abrupt interference, although the experiments do not establish this mechanism causally.

4.4 RQ3: Spectrum-Aware Structural Mitigation

We next evaluate whether concentrating directional adaptation in the low-energy spectral tail improves the observed retention–adaptation trade-off.

UIOrthoLoRA provides its clearest advantage in the small-capacity regime. On Gemma-3-12B-IT adapted on HotpotQA, among small-capacity configurations reaching adaptation accuracy of at least 0.98, UIOrthoLoRA attains the lowest Negative Shift ($1{,}548$, or 32.8% relative to $|\mathcal{D}_{\text{known}}|$), compared with DoRA ($1{,}571$; 33.3%), LoRA ($1{,}589$; 33.6%), and VeRA ($1{,}624$; 34.4%), while maintaining competitive external retention. We note that these margins are narrow: the span across the four methods is 76, or 1.6% relative to the known set, and each number comes from a single training run, so we read this ordering as an observed tendency rather than a statistically separated result.

The same pattern is particularly strong for Llama-3.2-3B-Instruct adapted on TriviaQA. Small UIOrthoLoRA reaches adaptation accuracy 0.935 while retaining an MMLU score of 0.549, compared with 0.342 for LoRA, 0.422 for DoRA, and 0.420 for VeRA. It also reaches a Negative Shift of $10{,}088$ (23.1% relative to $|\mathcal{D}_{\text{known}}|$), substantially lower than LoRA ($13{,}597$; 31.1%), DoRA ($13{,}910$; 31.8%), and VeRA ($13{,}805$; 31.6%). These small-tier configurations sit below the 0.98 adaptation gate used in the capacity figures, which is why the corresponding comparison in Appendix F shows no small tier for this setting.

At medium and large capacities, LoRA and DoRA become more competitive and can equal or exceed UIOrthoLoRA on individual retention metrics. UIOrthoLoRA therefore does not dominate

every method in every configuration. Its strongest evidence lies in the small-capacity results and in its smoother degradation across learning rates.

These findings answer RQ3 within the evaluated settings: restricting the most flexible directional adaptation to the low-energy spectral tail can improve retention while maintaining competitive target adaptation, particularly under constrained parameter budgets. The results do not establish superiority over retention-aware methods that were not included in the empirical comparison.

5 Conclusion

We studied how parameter-efficient fine-tuning changes outputs that an instruction-tuned language model previously generated correctly and reliably. We introduced a dual-axis evaluation framework that measures target adaptation and retention from repeated generations, together with Negative Shift, which identifies severe collapses in previously reliable generation.

Our experiments show that retention depends strongly on optimization, capacity, and adapter structure. For RQ1, the learning rate has a pronounced effect on the retention–adaptation tradeoff: aggressive optimization can sharply increase the Negative Shift and reduce the external benchmark performance. Among configurations achieving a comparable target adaptation, moving from small to medium capacity improves retention. For RQ2, LoRA and DoRA exhibit relatively abrupt degradation at aggressive learning rates, whereas VeRA and UIOrthoLoRA follow smoother retention trajectories, indicating that the adapter parameterization affects not only how much behavior is disrupted, but also how that disruption develops during adaptation.

For RQ3, UIOrthoLoRA provides its clearest advantage under constrained parameter budgets. By preserving the dominant pre-trained singular bases, it improves retention in the small-capacity settings while maintaining competitive target adaptation. At larger capacities, the baselines become more competitive, and no method dominates every retention measure.

Overall, our results show that an adapted model should be evaluated by its generation reliability. Generation-level retention analysis provides a direct way to expose such failures and to compare adapter configurations that achieve similar target performance.

Acknowledgments

We thank Orr Zwebner for helpful discussions and revisions of the paper.

This study is supported in part by the European Research Council (Intellexus, Project No. 101118558). Views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Limitations

Several limitations should be considered when interpreting our study.

Adaptation is measured on the training targets. The adaptation axis is evaluated on the same error-set examples used to train each adapter, so it measures whether the model learns to generate the target answers rather than generalization to held-out target questions.

Dependence on sampling and decoding. Generation scores are estimated from ten sampled generations under a fixed prompting and decoding configuration (temperature, top-p, and prompt format). A different configuration may yield different reliability estimates, so the scores should be read relative to this protocol.

Single run per configuration. Each reported configuration corresponds to one training run (all initializations use a fixed global seed; Appendix C), and we report no variance estimates across training seeds and perform no statistical power analysis (Miller, 2024). Small differences between methods, on the order of a few dozen in Negative Shift on known sets of several thousand prompts, should therefore be read as single-run observations rather than statistically separated results. The larger effects we emphasize, such as the abrupt learning-rate degradation of LoRA and DoRA and Negative Shift gaps of several thousand, are less likely to reflect run-to-run variation, but we did not verify this with repeated runs.

Scoring by alias matching. Correctness is determined by normalized matching against reference-answer aliases, using the rules specified in Appendix B. The matcher can err in both directions. It may miss semantically valid answers phrased differently (for example, a numeric quantity written

in words rather than digits), and its token-overlap and fuzzy rules may accept incorrect responses that share most tokens with a reference answer, such as a response naming only part of a multi-part answer or a near-identical but distinct entity name. The scores therefore approximate rather than perfectly capture correct generation. Two properties of the protocol limit, but do not remove, the effect of matcher errors on our comparisons: membership in $\mathcal{D}_{\text{known}}$ requires all ten base-model generations to pass the matcher, so a spuriously accepted answer must be produced consistently to contaminate the known set, and the same matcher scores generations before and after adaptation, so Negative Shift measures a change in matcher-judged behavior under identical acceptance rules. Re-scoring the framework under stricter matching, such as exact match only, is a robustness check we consider valuable future work.

Negative Shift threshold. Negative Shift uses $\tau = 0.8$ to emphasize severe collapses in the correct-generation rate; smaller but meaningful degradations do not enter the aggregate, and the threshold is interpretable but not universally optimal.

Scope of evaluation. Our study covers English factual question answering and selected external reasoning benchmarks. Long-form generation, dialogue, multilingual, and code settings are not evaluated, and benchmark retention does not guarantee preservation of all capabilities in deployment.

UIOrthoLoRA overhead. UIOrthoLoRA computes a one-time SVD of each adapted weight matrix and stores the resulting spectral factors, adding initialization cost and memory relative to standard low-rank adapters. It also introduces two structural hyperparameters ($k_{\text{val}}, k_{\text{vec}}$) whose optimal setting may depend on the model and target domain; we use static values and leave dynamic or learned allocation to future work.

Baseline scope. Our empirical comparison focuses on LoRA, DoRA, and VeRA because they represent learned low-rank directions, magnitude–direction decomposition, and learned scaling over frozen random projections. We do not compare directly against all retention-aware initialization and subspace methods discussed in Section 2. The results should therefore be interpreted as a comparison among the evaluated mechanisms, not as evidence of superiority over all retention-aware PEFT methods.

Risks. Adaptation can silently degrade outputs that were previously reliable while leaving overall fluency intact. In deployment, this may create misplaced confidence that a specialized model retains all capabilities of its base model. Generation-level regression evaluation can help expose such failures, although it does not guarantee reliability outside the evaluated prompts and decoding settings.

Supplementary Material Statement

Appendices A and C specify the complete inference and training configurations, including the prompting and decoding protocol, per-setting learning-rate grids, and optimizer settings; Appendix B specifies the full answer-matching procedure. Appendices D and H detail the UIOrthoLoRA implementation and algorithm. Appendices E–G report the full learning-rate sweeps, the capacity comparisons for all model–dataset settings, and the Negative Shift threshold-sensitivity analysis.

Data Availability Statement

TriviaQA, HotpotQA, MMLU, and BIG-bench are existing, publicly available datasets. Our implementation, the model-specific error and known splits, per-prompt pre-adaptation and post-adaptation generation scores, and the complete learning-rate and capacity sweep results are available at `https://github.com/Guy-Bilitski/what-did-my-adapter-break`.

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, Online. Association for Computational Linguistics.

AI @ Meta. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Klaudia Bałazy, Mohammadreza Banaei, Karl Aberer, and Jacek Tabor. 2024. LoRA-XS: Low-rank adaptation with extremely small number of parameters. *Preprint*, arXiv:2405.17604.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan

Frankle, Cody Blakeney, and John P. Cunningham. 2024. LoRA learns less and forgets less. *Transactions on Machine Learning Research*.

Gemma Team. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Dawid J Kopiczko, Tijmen Blankevoort, and Yuki M Asano. 2024. VeRA: Vector-based random matrix adaptation. In *The Twelfth International Conference on Learning Representations*.

Suhas Kotha, Jacob Mitchell Springer, and Aditi Raghunathan. 2024. Understanding catastrophic forgetting in language models via implicit inference. In *The Twelfth International Conference on Learning Representations*.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. 2024. DoRA: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*.

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

Minrui Luo, Fuhang Kuang, Yu Wang, Zirui Liu, and Tianxing He. 2025. SC-LoRA: Balancing efficient fine-tuning and knowledge preservation via subspace-constrained LoRA. *arXiv preprint arXiv:2505.23724*.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. 2024. PiSSA: Principal singular values and singular vectors adaptation of large language models. *Advances in Neural Information Processing Systems*, 37:121038–121072.

Evan Miller. 2024. Adding error bars to evals: A statistical approach to language model evaluations. *arXiv preprint arXiv:2411.00640*.

Sergey Pletenev, Maria Marina, Daniil Moskovskiy, Vasily Konovalov, Pavel Braslavski, Alexander Panchenko, and Mikhail Salnikov. 2025. How much knowledge can you pack into a LoRA adapter without harming LLM? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4309–4322.

Pratyusha Sharma, Jordan T Ash, and Dipendra Misra. 2024. The truth is in there: Improving reasoning in language models with layer-selective rank reduction. In *The Twelfth International Conference on Learning Representations*.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, and 1 others. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Pengwei Tang, Xiaolin Hu, Yong Liu, Lizhong Ding, Dongjie Zhang, Xing Wu, and Debing Zhang. 2026. Put the space of LoRA initialization to the extreme to preserve pre-trained knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 33232–33240. Association for the Advancement of Artificial Intelligence.

Jeffrey Uhlmann. 2018. A generalized matrix inverse that is consistent with respect to diagonal transformations. *SIAM Journal on Matrix Analysis and Applications*, 39(2):781–800.

Hanqing Wang, Yixia Li, Shuo Wang, Guanhua Chen, and Yun Chen. 2025. MiLoRA: Harnessing minor singular components for parameter-efficient LLM finetuning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4823–4836, Albuquerque, New Mexico. Association for Computational Linguistics.

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

Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen Leon Song, Jianlong Wu, Liqiang Nie, and Bernard

Ghanem. 2024. 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.

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

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*.

A Inference Details

To ensure reproducibility across both intrinsic (knowledge injection) and extrinsic (reasoning) evaluations, all inferences were conducted with the Language Model Evaluation Harness (`lm_eval`[1]) using bfloat16 precision.

Intrinsic Evaluation (Targeted QA). For the TriviaQA and HotpotQA evaluations, we followed the repeated-generation probing protocol of Pletenev et al. (2025) in a 5-shot setting (`num_fewshot=5`). For each prompt, we generated $N = 10$ independently sampled responses under the same prompt and decoding configuration. The generated responses were scored with the answer-matching procedure specified in Appendix B. The system prompt was:

> *"You are a trivia assistant. Answer each question with only the entity name, nothing else."*

We formatted five demonstrations as user-assistant turns (e.g., *"What is the capital of France?"* $\rightarrow$ *"Paris"*) with `temperature=0.5`, `top_p=0.9`, and `max_new_tokens=20`.

Extrinsic Evaluation (MMLU & BIG-bench). Broad-spectrum evaluation was conducted in the following way:

- **MMLU:** Evaluated in a zero-shot setting (`num_fewshot=0`) to isolate the model's zero-shot retrieval of world knowledge.
- **BIG-bench-9:** Evaluated across nine selected BIG-bench multiple-choice tasks: *Analytic Entailment, Cause and Effect, Conceptual Combinations, Causal Judgment, Analogical Similarity, Common Morpheme, Logical Deduction, Logical Sequence,* and *Odd One Out*.

[1] https://github.com/EleutherAI/lm-evaluation-harness

B Answer Matching Procedure

This appendix specifies the procedure that decides whether a generated response counts as correct in Equation 1. It documents the exact implementation used to produce all reported scores; the same procedure is applied to every generation, before and after adaptation, and is included in the released code.

Reference answers. For TriviaQA, $\mathcal{A}(x)$ contains the alias list distributed with the dataset (the `normalized_aliases` field together with the normalized answer value), which enumerates accepted surface forms of the same answer. For HotpotQA, the dataset provides a single gold answer string, so $\mathcal{A}(x)$ has one entry.

Response extraction. Each generated response is stripped of surrounding whitespace, and a leading "`answer:`" prefix is removed when present (case-insensitive). No other truncation is applied, so the full generated string (up to `max_new_tokens=20`) enters the matcher.

Normalization. The response and each reference answer are normalized identically before comparison: Unicode accents are removed (NFD decomposition, dropping nonspacing combining marks, Unicode category Mn), text is lowercased, each punctuation character is replaced by a space, the articles "a", "an", and "the" are removed, and whitespace is collapsed.

Matching rules. A response is counted as correct if it matches any reference answer in $\mathcal{A}(x)$. For each reference answer, the following rules are applied in order:

1. **Digit gate.** Numbers are extracted from the raw (pre-normalization) strings using the pattern `-?\d+(?:,\d+)*(?:\.\d+)?`. If either string contains numbers, the two extracted number sets must be exactly equal; otherwise this reference answer is rejected without applying the remaining rules.
2. **Exact match.** The normalized strings are equal.

3. **Token overlap.** Treating each normalized string as a set of tokens, and provided both sets are non-empty: the response tokens are a subset of the reference tokens; or the reference tokens are a subset of the response tokens and the response contains at most two additional tokens.

4. **Fuzzy match.** The `rapidfuzz` `token_sort_ratio` between the two normalized strings is at least 85.

Properties and consequences. The digit gate makes numeric answers strict, and strict in a literal sense: a response and a reference expressing the same quantity in different forms ("12 thousand" versus "12,000") do not match. The token-overlap and fuzzy rules are deliberately permissive toward word order and short paraphrases, which reduces false negatives on entity answers but can also accept responses a stricter matcher would reject, for example a response naming only one part of a multi-part answer, or a near-identical but distinct name that clears the fuzzy threshold. The Limitations section discusses how these error modes bear on our measurements; the central mitigating property is that the identical matcher scores pre-adaptation and post-adaptation generations, so every comparison in the paper is between quantities measured under the same acceptance rules.

C Training Details

All parameter-efficient fine-tuning was executed using the Hugging Face `trl SFTTrainer`[2].

Optimization Setup. Models were trained for 10 epochs using the Fused AdamW optimizer. We applied a cosine learning-rate scheduler with a warmup ratio of 0.05. The maximum gradient norm was clipped at 1.0. Training was conducted with a per-device batch size of 64 and a maximum sequence length of 1024 tokens. All training utilized bfloat16 precision to match the base models.

Adapter Configurations and Sweeps. All baseline adapters were applied comprehensively across all linear layers in the attention and feed-forward blocks (`q_proj`, `v_proj`, `k_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`). The LoRA scaling scalar α was fixed at 32, and dropout was disabled (0.0).

[2] https://huggingface.co/docs/trl/sft_trainer

To rigorously establish the optimal Pareto frontier for all adapters, we swept learning rates spanning 10^{-6} to 10^{-2}.

- **Gemma-3-12B-IT, HotpotQA:** 10^{-6}, 5×10^{-6}, 10^{-5}, 5×10^{-5}, 10^{-4}, 5×10^{-4}, 10^{-3}, 5×10^{-3}, 10^{-2};
- **Gemma-3-12B-IT, TriviaQA:** 5×10^{-6}, 10^{-5}, 5×10^{-5}, 10^{-4}, 5×10^{-4}, 10^{-3}, 5×10^{-3}, 10^{-2};
- **Llama-3.2-3B-Instruct, HotpotQA:** 5×10^{-6}, 10^{-5}, 5×10^{-5}, 10^{-4}, 5×10^{-4}, 10^{-3}, 5×10^{-3}, 10^{-2};
- **Llama-3.2-3B-Instruct, TriviaQA:** 10^{-5}, 5×10^{-5}, 10^{-4}, 5×10^{-4}, 10^{-3}, 5×10^{-3}, 10^{-2}.

All randomized initializations (e.g., standard LoRA matrices or orthogonal rotations) were anchored to a global seed of 42 to ensure deterministic initialization across runs.

D UIOrthoLoRA Implementation Details

D.1 Forward Pass and Parameter Efficiency

During training we avoid materializing the dense update matrix ΔW and instead apply the adapter sequentially. We use the row-vector batch convention $X \in \mathbb{R}^{B\times L\times d_{\text{in}}}$, for which the adapter branch computes $X(\Delta W)^\top$ with ΔW as in Equation 11:

$$\begin{aligned} X \xrightarrow{\odot D} X_{\text{sc}} \xrightarrow{\times \widetilde{V}} H_{\text{proj}} \xrightarrow{\odot \widetilde{\alpha}} H_{\text{scale}} \\ \xrightarrow{\times \widetilde{U}^\top} Y_{\text{out}} \xrightarrow{\odot E} Y_{\text{final}} \end{aligned} \tag{13}$$

Here $D \in \mathbb{R}^{d_{\text{in}}}$ scales the input features, $\widetilde{V}$ projects them onto the (partly rotated) right singular directions, the coefficients $\widetilde{\alpha}$ scale the latent activations, $\widetilde{U}^\top$ maps back to the output space, and $E \in \mathbb{R}^{d_{\text{out}}}$ applies output scaling; the result is added to the frozen base output $XW_0^\top$.

The trainable parameters are the diagonal scalers D and E, the coefficients α_{mid} and α_{tail}, and two orthogonally parametrized matrices $R_U, R_V \in \mathrm{O}(k_{\text{vec}})$, implemented using `torch.nn.utils.parametrizations.orthogonal`. This keeps the number of trainable parameters comparable to standard low-rank adapters while leaving the dominant singular bases fixed. The coefficients α_{mid} and α_{tail} are initialized to the corresponding singular values of W_0, and the scalers are initialized to $D = \mathbf{0.1}$ and $E = \mathbf{0.1}$.

D.2 Efficient Spectral Initialization

A naive implementation that recomputed the SVD at every forward pass would be prohibitive

($\mathcal{O}(\min(mn^2, m^2n))$). We instead perform the decomposition once. At initialization we compute a single SVD of the pre-trained weight W_0, partition the resulting factors U, σ, and $V^\top$ into the dominant, intermediate, and tail regions of Section 3.2, and register them as frozen, non-trainable factors that are saved with the adapter checkpoint and receive no gradient updates. The SVD is not recomputed during training: each adapter update reuses these frozen factors together with the trainable coefficients $\alpha_{\text{mid}}, \alpha_{\text{tail}}$ and rotations R_U, R_V. The UIOrthoLoRA algorithm is given in Algorithm 1 (Appendix H).

E Learning-Rate Sweeps Across Model–Dataset Settings

Figures 4–6 give the complete per-configuration learning-rate sweeps for the three model–dataset settings not shown in the main paper, using the same axes as Figure 2: target adaptation against Negative Shift (↓) and against external retention (unweighted mean of MMLU and BIG-bench-9 accuracy, ↑). Each point is one trained configuration and color encodes the learning rate; the per-setting learning-rate grids are listed in Appendix C.

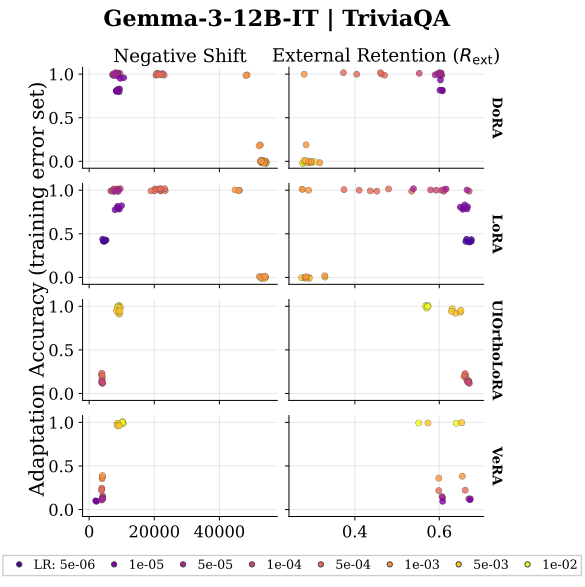


Figure 4: Learning-rate sweep for Gemma-3-12B-IT adapted on TriviaQA.

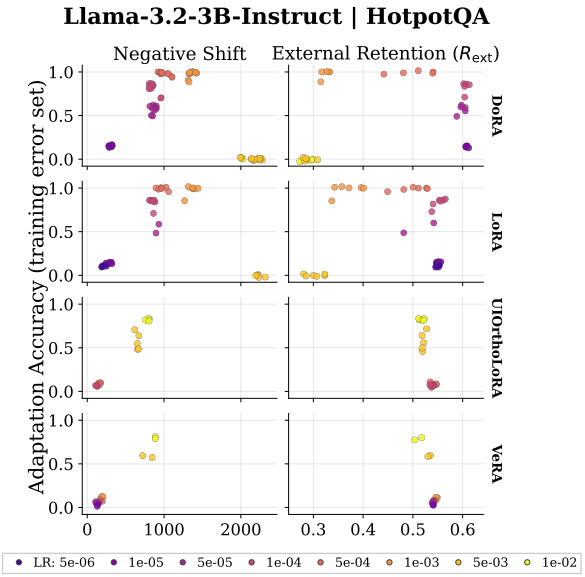


Figure 5: Learning-rate sweep for Llama-3.2-3B-Instruct adapted on HotpotQA.

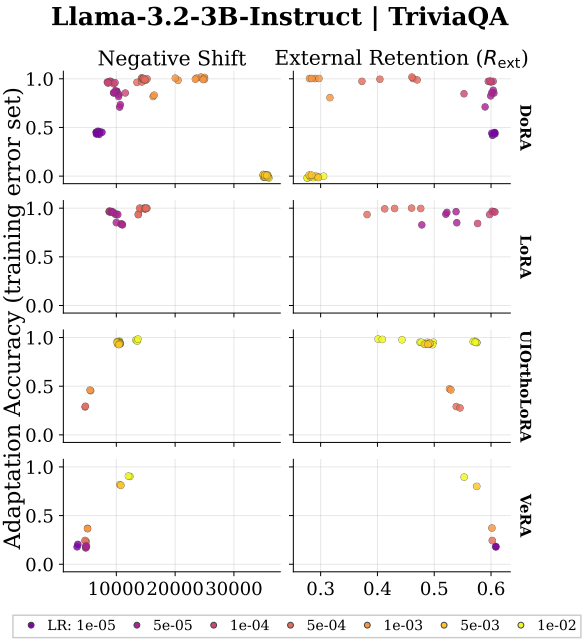


Figure 6: Learning-rate sweep for Llama-3.2-3B-Instruct adapted on TriviaQA.

F Capacity Comparisons Among Matched-Adaptation Configurations

Figures 7–9 give the capacity comparisons for the three model–dataset settings not shown in the main paper, using the same convention as Figure 3: only configurations meeting the adaptation threshold (accuracy ≥ 0.98) and having complete external evaluations are displayed, grouped by capacity tier (small, medium, large; by trainable parameter count). A bar is absent when no configuration

in that method–tier met the adaptation threshold; this is distinct from tiers that were not evaluated (VeRA, whose capacity is effectively fixed, appears only in the small tier and only where a qualifying small-capacity configuration exists). No bars are added for configurations that did not meet the threshold. Concretely, for Llama-3.2-3B-Instruct on HotpotQA (Figure 8) only UIOrthoLoRA has a small-tier configuration that meets the threshold, and on TriviaQA (Figure 9) no small-tier configuration of any method reaches it, so the small tier is absent from that comparison. Small-tier configurations for these settings do exist in the sweep but remain below the 0.98 gate; Table 2 reports representative small-tier runs selected without this gate.

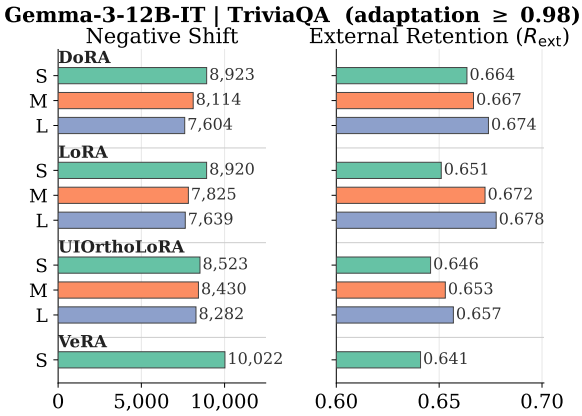


Figure 7: Matched-adaptation capacity comparison for Gemma-3-12B-IT adapted on TriviaQA.

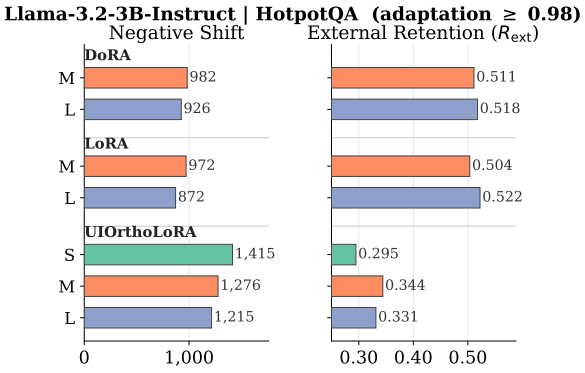


Figure 8: Matched-adaptation capacity comparison for Llama-3.2-3B-Instruct adapted on HotpotQA. Only UIOrthoLoRA has a small-tier configuration meeting the ≥ 0.98 adaptation threshold in this setting.

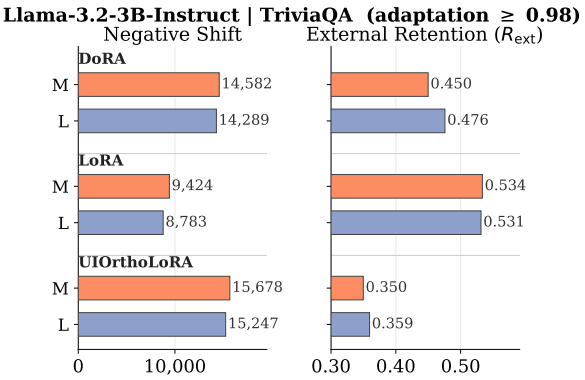


Figure 9: Matched-adaptation capacity comparison for Llama-3.2-3B-Instruct adapted on TriviaQA. No small-tier configuration of any method meets the ≥ 0.98 adaptation threshold in this setting, so the small tier is absent.

G Sensitivity to the Negative Shift Threshold

The main paper uses a Negative Shift threshold of $\tau = 0.8$ (Equation 5), which admits only severe collapses in the correct-generation rate. To verify that our comparative conclusions are not an artifact of this specific value, we recomputed Negative Shift under a more lenient alternative threshold, $\tau = 0.6$, for every trained configuration.

As expected, Negative Shift increases at the lower threshold, because $\tau = 0.6$ additionally admits known-set prompts whose correct-generation rate falls from 1.0 to as high as 0.4. The comparative structure, however, is preserved: across all four model–dataset settings, the per-configuration Negative Shift values at $\tau = 0.6$ and $\tau = 0.8$ are strongly rank-correlated (Spearman ρ between 0.93 and 0.98, over up to 194 configurations per setting). The comparative patterns across adapter mechanisms, learning rates, and capacity tiers are therefore similar under both thresholds, and the main-paper conclusions do not depend on the choice $\tau = 0.8$. We emphasize that this is threshold *robustness* (similar comparative conclusions) rather than identical values, which necessarily change with τ.

Table 3 illustrates this stability for a representative learning-rate sweep. Both thresholds trace the same trajectory (moderate degradation at small learning rates, rising through the mid-range, and a sharp increase at aggressive learning rates where target adaptation itself collapses), with $\tau = 0.6$ shifted upward by a roughly constant offset.

Learning rate	Adaptation	NS (τ=0.8)	NS (τ=0.6)
10^{-5}	0.306	1,142	1,481
5×10^{-5}	0.967	1,573	1,732
10^{-4}	0.993	1,554	1,729
5×10^{-4}	0.997	1,943	2,140
10^{-3}	0.996	2,899	3,077
5×10^{-3}	0.023	3,874	4,094
10^{-2}	0.007	4,069	4,307

Table 3: Negative Shift under the main threshold $\tau = 0.8$ and the alternative $\tau = 0.6$, for a representative DoRA configuration (rank 16) on Gemma-3-12B-IT adapted on HotpotQA, across learning rates from 10^{-5} to 10^{-2}. Values rise at the more lenient $\tau = 0.6$, but both thresholds trace the same trajectory, so the comparative trend is preserved.

H UIOrthoLoRA Algorithm

Algorithm 1 UIOrthoLoRA: Frozen Spectral Initialization & Forward Pass

Require: Base weight $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, spectral dimension $r = \min(d_{\text{out}}, d_{\text{in}})$

Require: Hyperparameters: k_{val} (trainable adapter coefficients), k_{vec} (adaptable tail directions)

Initialization (run once):

$U, \sigma, V^\top \leftarrow \text{SVD}(W_0)$

$r_{\text{frozen}} \leftarrow r - k_{\text{val}}$; $\quad r_{\text{stable}} \leftarrow r - k_{\text{vec}}$

Slice and freeze spectral bases

$U_{\text{fix}} \leftarrow U_{:,:r_{\text{stable}}}$

$V_{\text{fix}}^\top \leftarrow V_{:r_{\text{stable}},:}^\top$

$\alpha_{\text{fix}} \leftarrow \mathbf{1} \in \mathbb{R}^{r_{\text{frozen}}}$ {Dominant adapter coefficients fixed to one}

Initialize trainable parameters

$\alpha \in \mathbb{R}^{k_{\text{val}}} \leftarrow \text{init}(\sigma_{r_{\text{frozen}}:})$ {$\alpha = [\alpha_{\text{mid}}; \alpha_{\text{tail}}]$}

Initialize trainable square matrices $R_U, R_V \in \mathbb{R}^{k_{\text{vec}} \times k_{\text{vec}}}$

Register orthogonal parametrizations on R_U and R_V {`torch.nn.utils.parametrizations.orthogonal`}

Initialize $D \in \mathbb{R}^{d_{\text{in}}}$ and $E \in \mathbb{R}^{d_{\text{out}}}$ {Input and output feature scalers}

Forward pass (input $x \in \mathbb{R}^{d_{\text{in}}}$):

1. Construct the adaptive tail bases

$\hat{U}_{\text{adapt}} \leftarrow U_{:,r_{\text{stable}}:} R_U$

$\hat{V}_{\text{adapt}}^\top \leftarrow R_V^\top V_{r_{\text{stable}}:,:}^\top$

2. Concatenate fixed and adaptive components

$\hat{U} \leftarrow [U_{\text{fix}}, \hat{U}_{\text{adapt}}]$

$\hat{\alpha} \leftarrow [\alpha_{\text{fix}}; \alpha]$

$\hat{V}^\top \leftarrow [V_{\text{fix}}^\top; \hat{V}_{\text{adapt}}^\top]$

3. Compute the adapter output

$h \leftarrow \hat{V}^\top (x \odot D)$ {Project onto spectral coordinates}

$y \leftarrow \hat{U}(h \odot \hat{\alpha})$ {Apply coefficients and map to output space}

return $W_0 x + (y \odot E)$ {Residual connection}

I AI Assistants

We utilized AI assistants, such as ChatGPT, to assist with code formatting, phrasing suggestions, and LaTeX styling during writing. All outputs were reviewed and edited by the authors, ensuring no content was generated or used without human

verification.