BEYOND PERPLEXITY: A GEOMETRIC AND SPECTRAL STUDY OF LOW-RANK PRE-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Pre-training large language models is dominated by the memory cost of storing full-rank weights, gradients, and optimizer states. Low-rank pre-training has emerged to address this, and the space of methods has grown rapidly. A central question remains open: do low-rank methods produce models that generalize comparably to full-rank training, or does the rank constraint fundamentally alter the solutions reached? Existing comparisons rely almost entirely on validation perplexity from single-seed runs, often carried forward from prior literature. Yet perplexity is a poor proxy for solution quality — two methods can match on perplexity while converging to very different loss landscape regions and internal representations. To look beyond perplexity, we introduce **BASIN (Barrier, Activations, Spectra, and Intrinsic geometry of Networks)**, a framework that characterizes the solutions found by five low-rank pre-training methods — GaLore and Fira (memory-efficient optimizers), CoLA and SLTrain (architecture reparameterizations), and ReLoRA (adapter-style updates with periodic resets) — against full-rank training at three model scales (60M, 130M, 350M) under three seeds. We evaluate each along 16 metrics across four dimensions: 1-D loss landscape along random/top-K PCA directions, 1-D interpolation between checkpoints, spectral structure of the weights and learned updates, and activation similarity to full-rank training. We also examine how each pre-trained model adapts after task and instruction fine-tuning, and continual pre-training with and without replay. We show that low-rank methods are not equivalent to full-rank training, nor to one another, even when their validation perplexity is close. Each method converges to a distinct basin with different sharpness and spectrum. Low-rank method activations drift further from full-rank in later layers and as training progresses. These differences shape post-training behavior. For example, the path from the pre-trained to the fine-tuned model crosses a barrier for ReLoRA and climbs a steep wall for SLTrain, matching their high forgetting. Finally, in two case studies, BASIN traces how Fira and SLTrain deviate from full-rank, and a simple fix guided by this diagnosis brings each method closer to full-rank training.

1 INTRODUCTION

Pre-training large language models (LLMs) is dominated by the memory cost of storing full-rank weights, gradients, and optimizer states. Low-rank pre-training has emerged as a way to reduce this cost while still approximating full-rank training. This field has evolved along three main directions. Optimizer-based methods such as GaLore (Zhao et al., 2024) leave the model intact and constrain only the gradient update to a low-rank subspace. Architecture-based methods such as CoLA (Liu et al., 2025b) replace dense linear layers with low-rank decompositions. Adapter-based methods such as ReLoRA (Lialin et al., 2024) add low-rank adapters to linear layers that are periodically merged into the weights and re-initialized.

Despite this growing body of work, evaluation practices in low-rank pre-training remain limited. Methods are benchmarked against one another using validation perplexity from a single training run with a single random seed (Zhao et al., 2024; Lialin et al., 2024; Liu et al., 2025b; Han et al., 2024; Loeschcke et al., 2024; Zhou et al., 2024; Muhamed et al., 2024; Shivagunde et al., 2024; Li et al., 2025; Miles et al., 2024). While this is understandable given the significant compute costs of large-scale pre-training, it raises an important question: is validation perplexity alone a sufficient

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

criterion for evaluating a low-rank pre-training method's quality? Earlier work Liu et al. (2023) shows that two models with the same pre-training loss can differ substantially on downstream tasks, and that loss-landscape flatness predicts this gap. Beyond validation perplexity, most prior work evaluates low-rank pre-training methods by using them as fine-tuning recipes. For example, GaLore (Zhao et al., 2024), SLTrain (Han et al., 2024), LOST (Li et al., 2025), and VeLoRA (Miles et al., 2024) are evaluated by fine-tuning RoBERTa on GLUE, while CoLA (Liu et al., 2025b) is evaluated by fine-tuning LLaMA-7B on downstream tasks. This measures how well a low-rank method fine-tunes a pre-trained model, but does not tell us how a model pre-trained with the low-rank method generalizes downstream. A few studies do fine-tune pre-trained models directly. For instance, Lialin et al. (2024) compare ReLoRA against full-rank on GLUE, (Liu et al., 2025b) compare a pre-trained CoLA against BERT, and Zhou et al. (2024) compare full-rank, GaLore, and SwitchLoRA. These studies are narrow in scope, covering one or two methods at a single scale, and none span all three method families (optimizers, model architecture, adapters).

Only a handful of recent studies look beyond validation perplexity, and these are narrow in scope. FlatLoRA (Li et al., 2024) compares LoRA against full-rank training via loss landscape analysis, but only in the fine-tuning setting. Liu et al. (2025a) study GaLore, LoRA, and full-rank training through the Polyak-Łojasiewicz (PL) condition, but their analysis is restricted to multi-layer perceptrons and does not transfer cleanly to transformers. Both studies cover only a subset of low-rank methods and rely on a single analytical lens.

We address this research gap with five contributions: **(1) BASIN: A diagnostic framework for low-rank pre-training**, including a novel cross-method barrier metric (IMBH) that quantifies geometric separation between training algorithms — not just between seeds or epochs as in prior linear-mode-connectivity work. **(2) Cross-family empirical findings** that no prior work establishes: low-rank methods occupy geometrically separated basins, develop distinct spectral signatures, and produce divergent internal representations. **(3) Post-training behavior of low-rank methods.** Beyond zero-shot evaluation, we evaluate each method in four post-training settings: task fine-tuning, instruction fine-tuning, and continued pre-training on German with and without replay. **(4) From diagnosis to design.** BASIN explains post-training behavior: methods whose path to the fine-tuned model crosses a barrier or climbs a steep wall (ReLoRA, SLTrain) forget the most. In two case studies, BASIN traces how Fira and SLTrain deviate from full-rank, and a simple fix guided by this diagnosis brings each method closer to full-rank training in both geometry and post-training behavior. **(5) An open-source unified codebase.** We release a single pipeline integrating pre-training, post-training, metric computation, downstream evaluation, designed for easy extension to new methods and metrics, enabling the community to study low-rank pre-training in more depth and design better methods.

2 Related Work

Low-rank pre-training methods. Memory-efficient pre-training of LLMs has motivated a fast-growing family of low-rank methods, which broadly fall into three design categories. The first category wraps the optimizer, constraining the gradient update to a low-rank subspace while leaving the parameters at full rank. GaLore (Zhao et al., 2024) projects gradients into a low-rank subspace, applies an Adam update there, and projects back to the full-rank parameter space. Fira (Chen et al., 2024) introduces a norm-based scaling that uses GaLore's low-rank Adam states to scale the full-rank gradient, recovering full-rank update directions at the same memory cost. A growing line of work extends this design idea along several axes, including Q-GaLore (Zhang et al., 2024), GaLore 2 (Su et al., 2025), GRASS (Muhamed et al., 2024), GaLore-mini (Huang et al., 2024), COAP (Xiao et al., 2025), and FLORA (Hao et al., 2024). The second category modifies the model itself, replacing dense weight matrices with low-rank decompositions. CoLA (Liu et al., 2025b) restructures attention and MLP linear layers as bottleneck low-rank matrices with a non-linearity in between. SLTrain (Han et al., 2024) parameterizes weights as a sparse-plus-low-rank sum, jointly trained from scratch. ReDoRA and RePiSSA (Shivagunde et al., 2024) extend the same idea with alternative reparameterization choices. The third category adds low-rank adapters that are periodically merged back and re-initialized. ReLoRA (Lialin et al., 2024), sparked the initial interest in low-rank pre-training; SwitchLoRA (Zhou et al., 2024) extends it by periodically switching parameters of the LoRA matrices, and LoQT (Loeschcke et al., 2024) adapts the recipe to quantized models.

Toward understanding the training dynamics of low-rank methods. A handful of studies probe low-rank training beyond final validation loss, but each examines a single signal in isolation. Flat-

LoRA (Li et al., 2024) and Liu et al. (2025a) take a loss-landscape view. SLTrain (Han et al., 2024) tracks singular-value spectra to argue that the sparse-plus-low-rank parameterization preserves rank that pure low-rank loses, ReLoRA (Lialin et al., 2024) inspects the rank of cumulative updates to argue that periodic merges grow effective rank beyond a single low-rank stage, and WeLore (JAISWAL et al.) characterizes the non-uniform low-rank structure of pre-trained LLM weights across layers and exploits it for fine-tuning.

Geometry- and spectrum-based analyses of trained models. Li et al. (2018) pioneers the visualization of the loss landscape itself and shows that design choices — such as architecture, optimizer, and batch size — shape the underlying loss landscape, and that some landscapes are far easier for an optimizer to navigate than others. Their study, however, was conducted on image classifiers, not LLMs. Subsequent work has used loss landscapes, weight spectra, and activation similarity to study different aspects of LLM training. For example, Wen et al. (2024) uses 1-D loss profiles to argue that LLM training under warmup-stable-decay schedules traces a "river valley" geometry. Chen et al. (2025) adapt random-direction perturbations to LLMs to characterize the basin-like structure of their loss landscape. Liu et al. (2023) uses loss-landscape to show that two language models reaching the same pre-training loss can differ on downstream tasks, with flatness predicting the gap where perplexity does not. However, Andriushchenko et al. (2023) and Kaur et al. (2023) show that sharpness alone correlates poorly with generalization, motivating us to include additional signals such as rank and activation metrics, which have been used to study other aspects of model behavior like model mergeability (Zhou et al., 2026).

3 BASIN: A DIAGNOSTIC FRAMEWORK FOR LOW-RANK METHODS

To look beyond perplexity, we introduce **BASIN** (Barrier, Activation, Spectra, and Intrinsic geometry of Networks), a new diagnostic framework that compares low-rank pre-training methods with full-rank training. BASIN has sixteen metrics grouped into four lenses: **B**arriers, 1-D interpolation between checkpoints; **A**ctivations, activation similarity to full-rank training; **S**pectra, spectral structure of the weights and learned updates; and **I**ntrinsic geometry of **N**etworks, the 1-D loss landscape along random and top-1 PCA direction. We apply it to full-rank, GaLore, Fira, CoLA, SLTrain, and ReLoRA at 60M, 130M, and 350M parameters across three seeds. Each lens asks primarily one question. **Barrier**: do methods land in the same basin? **Activation**: do they compute similar representations? **Spectra**: do their weights have similar rank structure? **Intrinsic geometry**: how sharp is the basin around each solution? We introduce each lens and its metrics below.

3.1 BARRIER VIA 1-D INTERPOLATION BARRIER HEIGHT

If two models lie in the same basin, the loss should stay low along the straight path between them. Otherwise, the path must cross a region of high loss, a barrier. We use this idea to measure barrier height between two checkpoints. Given two checkpoints θ_A and θ_B, we evaluate $\mathcal{L}(\beta) = \mathcal{L}_{\text{val}}\big((1-\beta)\,\theta_A + \beta\,\theta_B\big)$ for $\beta \in [0,1]$ and, following Frankle et al. (2020), measure the barrier height $\text{BH} = \max_{\beta\in(0,1)} \mathcal{L}(\beta) - \frac{1}{2}\big(\mathcal{L}(0) + \mathcal{L}(1)\big)$. A BH near zero means both models lie in the same basin, and a large BH means they lie in different basins. We compute BH in two settings. **Consecutive Checkpoints Barrier Height (CCBH)** interpolates between consecutive checkpoints of the same method (e.g., $\theta_{1\text{k}}$ and $\theta_{2\text{k}}$) throughout training. **Inter-Method Barrier Height (IMBH)** interpolates between checkpoints of different methods at the same step.

3.2 ACTIVATION METRICS.

To measure how closely a low-rank checkpoint's internal representations match those of the full-rank baseline, we compare their hidden states on identical inputs. We compute **Activation L2 distance**, which measures the mean per-position Euclidean distance between the two activation matrices. **Activation cosine similarity** reports the mean directional alignment of activation vectors at each position, isolating angular mismatch independent of magnitude. **Linear CKA** (Kornblith et al., 2019) (refer to Appendix C.1) compares the two activation matrices at the representational level and is invariant to orthogonal rotations and isotropic scaling.

3.3 SPECTRAL AND RANK METRICS.

Following Zhou et al. (2026), we track the spectral structure of parameters and their updates over the training period. We use four metrics: **effective rank**, the entropy of the normalized singular value spectrum, which measures how broadly energy spreads across directions; **stable rank**, the ratio of the Frobenius norm to the operator norm, a noise-robust estimate of dimensionality; **spectral gap**, the relative separation between the top two singular values, which shows whether one direction dominates; and **threshold rank**, the number of singular values above a fixed cutoff. We compute these on the materialized weights, so all methods share the same weight space, and on the updates between consecutive checkpoints. Refer to Appendix C.2 for metric definitions.

3.4 INTRINSIC GEOMETRY VIA 1-D LOSS LANDSCAPE

To probe the local geometry around each checkpoint θ, we follow Chen et al. (2025) and track the C4 validation loss $\mathcal{L}(\alpha) = \mathcal{L}_{\text{val}}(\theta + \alpha\,\delta)$ along $D = 100$ random directions $\delta_i \sim \mathcal{N}(0, I)$. We summarize each checkpoint by its **sharpness** $S = \frac{1}{2ND} \sum_{i,j} \big[\Delta(\delta_i, +\alpha_j) + \Delta(\delta_i, -\alpha_j)\big]$, the mean loss increase $\Delta(\delta, \alpha) = \mathcal{L}(\delta, \alpha) - \mathcal{L}(\delta, 0)$ over N offsets and both signs (larger S = sharper basin), and by its **direction variance**, the across-direction variance of $\mathcal{L}$, which measures anisotropy (Appendix F). Random directions probe average-case geometry, so we also perturb along each weight matrix's **top-k PCA direction** $\delta^{(k)} = \sigma_k u_k v_k^\top$, where $W = U\Sigma V^\top$.

4 EXPERIMENTAL SETUP

Pre-training: We use the codebase from Zhao et al. (2024) to pre-train LLaMA-style decoders at 60M, 130M, and 350M parameters on C4 dataset with Chinchilla-optimal token budgets with three seeds. We compare full-rank training with low-rank methods such as GaLore, Fira, CoLA, SLTrain, and ReLoRA without warm start. For the low-rank methods we use a rank of 128 for 60M models and a rank of 256 for 130M and 350M models keeping other hyperparameters from the respective method original papers. **Post-training.** We test each method's final pre-trained checkpoint in four post-training settings: task specific SFT on five benchmarks (ARC-Easy, HellaSwag, PIQA, SWAG, LogiQA2), instruction SFT on LaMini (Wu et al., 2024), and continued pre-training (CPT) on German C4, with 25% English replay and without replay, following Ibrahim et al. (2024). In both SFT settings, the fine-tuning recipe is fixed across methods, so the only variable is the pre-trained checkpoint. For task SFT, the learning rate and step budget are tuned once on the full-rank model per task and scale, then reused for all other methods. SFT is run at all three scales. CPT is run at 130M: each model continues training with its own pre-training method, on the same token budget as pre-training ($\approx$2.3B tokens). In every setting, we track three quantities: *target performance* (how well the model does on the new data), *gain* (improvement over the pre-trained checkpoint), and *forgetting* (increase in held-out English C4 validation loss). Full details are in Appendix B. **BASIN Metrics:** We compute all metrics in Section 3 on a fixed 1000-example C4 validation subset with a shared seed. **Downstream evaluation:** We evaluate final checkpoints zero-shot with lm-evaluation-harness (Gao et al., 2024) on 11 tasks across five categories: commonsense (HellaSwag, PIQA, COPA, SWAG), world knowledge (OpenBookQA, ARC-Easy, QA4MRE-2013), reading comprehension (ReCoRD), grammar (BLiMP), and logical reasoning (LogiQA, LogiQA2). All tasks score above random performance and show gains with model scale. We report mean and variance in Tables 5–7 in Appendix D. We use an aggregate plot across all tasks in Figure 2. All experiments were run on NVIDIA RTX 6000 Ada Generation GPUs (48 GB). Each training run and metric computation fits on a single GPU.

5 RESULTS

Section 5.1 shows that perplexity cannot predict post-training behavior. Section 5.2 applies BASIN to the pre-trained checkpoints, Section 5.3 uses it to explain post-training behavior, and Section 5.4 uses it to better align Fira and SLTrain with full-rank.

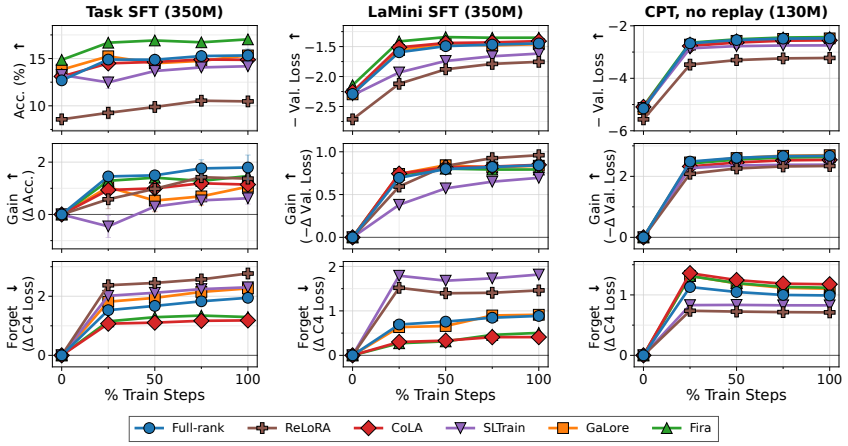


Figure 1: Post-training target performance, gain, and C4 forgetting (mean ± s.d., 3 seeds). SFT at 350M, CPT at 130M.

5.1 Validation perplexity cannot capture low-rank post-training behavior

Zero-shot evaluation: Table 1 shows validation perplexity. The ranking at 130m and 350m is: Fira < Full-rank < GaLore ≈ CoLA < SLTrain < ReLoRA. Fira beats full-rank at every scale (17.26 vs. 19.08 at 350M), and ReLoRA is far behind (29.30 at 350M). Figure 2 shows the zero-shot macro z-score over 11 tasks. **Only the two extremes keep their place: Fira is best and ReLoRA is worst at every scale. The four middle methods overlap** within standard deviation. For example, SLTrain has higher perplexity than full-rank at every scale (e.g., 20.66 vs. 19.08 at 350M), yet its zero-shot score cannot be told apart from full-rank. Zero-shot gives the order Fira > full-rank ≈ GaLore ≈ CoLA ≈ SLTrain > ReLoRA. Therefore, a gap in perplexity does not always show up downstream, and zero-shot alone cannot separate the middle methods.

Supervised fine-tuning (SFT). Figure 1 (first two columns) shows the 350M results for task SFT and LaMini SFT. **SFT separates methods that zero-shot could not, and the separation grows from task SFT to LaMini SFT.** In task SFT, SLTrain drops out of the middle group, with lower accuracy at two of three scales and more C4 forgetting at all scales, giving Fira > full-rank ≈ GaLore ≈ CoLA > SLTrain > ReLoRA. On LaMini, CoLA also breaks away and ranks second at two of three scales, giving Fira > CoLA > full-rank ≈ GaLore > SLTrain > ReLoRA. Across both settings and all scales, Fira reaches the best target performance and forgets less. SLTrain and ReLoRA are worst on both counts.

Continual pre-training (CPT). The last column of Figure 1 shows the CPT results at 130M. On German loss, the ranking is Fira ≈ full-rank ≈ GaLore > CoLA > SLTrain > ReLoRA. **Surprisingly, CPT reverses the forgetting pattern seen in SFT.** Fira comparitively gains more but also forgets more English. SLTrain and ReLoRA gain the least but also forget the least.

Takeaway: Low-rank methods behave differently under each post-training protocol, so evaluating them requires going beyond validation perplexity.

5.2 BASIN shows differences which perplexity misses

We now examine the pre-trained checkpoints through the four BASIN lenses. Plots show a single seed for clarity; every ordering we report holds beyond noise across three seeds.

Do low-rank methods converge to different basins? If they do, interpolating between two methods should cross a high loss barrier. Figure 3 left shows the CCBH for each method. **CCBH decays for all methods as training progresses.** Across scales and beyond noise, GaLore has the lowest CCBH and ReLoRA the highest. The order of the other methods is inconclusive. Figure 3 right shows the IMBH. Every method pair has a large barrier ($BH \approx 2$–8) at all scales and steps, so all methods occupy separate basins separated by a barrier (also see Figure 7b). Unlike CCBH, **IMBH grows with training: methods drift apart as each settles into its own solution basin.** At 60M, the barriers to

Method	60M	130M	350M
Full-rank	$\underline{34.04}^{b}_{\pm 0.08}$	$\underline{25.07}^{b}_{\pm 0.03}$	$\underline{19.08}^{b}_{\pm 0.35}$
GaLore	$34.87^{b}_{\pm 0.04}$	$25.52^{c}_{\pm 0.09}$	$19.88^{c}_{\pm 0.02}$
Fira	$\mathbf{31.04}^{a}_{\pm 0.08}$	$\mathbf{23.18}^{a}_{\pm 0.01}$	$\mathbf{17.26}^{a}_{\pm 0.12}$
CoLA	$34.28^{b}_{\pm 0.71}$	$25.52^{c}_{\pm 0.13}$	$19.85^{c}_{\pm 0.23}$
SLTrain	$35.68^{c}_{\pm 0.10}$	$26.61^{d}_{\pm 0.11}$	$20.66^{d}_{\pm 0.01}$
ReLoRA	$42.88^{d}_{\pm 0.11}$	$36.01^{e}_{\pm 0.16}$	$29.30^{e}_{\pm 0.61}$

Table 1: C4 validation perplexity (mean ± s.d. over three seeds; lower is better). Bold marks the best method and underline the second best at each scale. Superscript letters give the overall order of the methods. Methods that share a letter are tied.

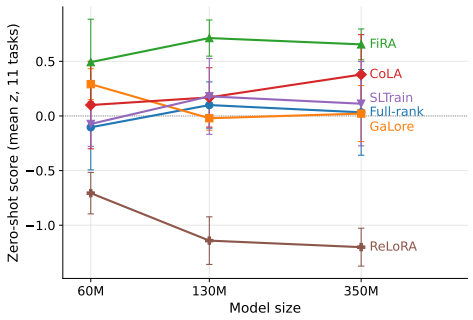


Figure 2: Zero-shot macro z-score over 11 tasks (higher is better; mean ± s.d. over three seeds). Only the extremes (Fira, ReLoRA) keep their perplexity rank.

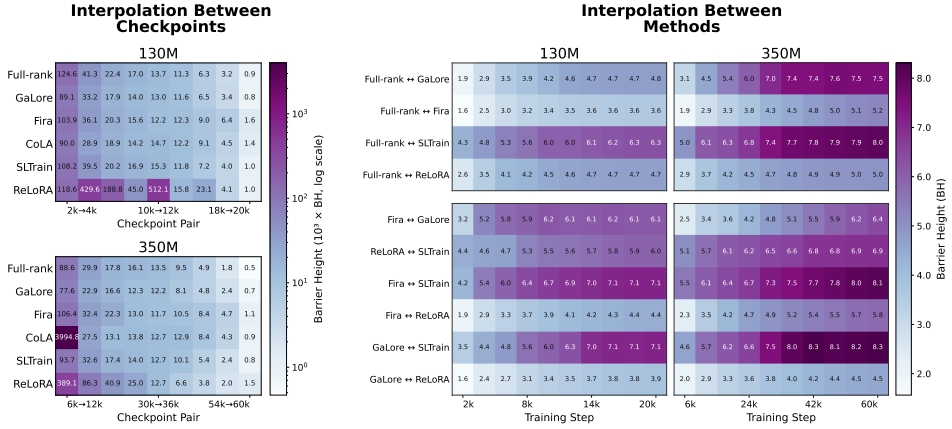


Figure 3: **(a)** CCBH. **(b)** IMBH. Darker means a higher barrier.

full-rank cannot be separated from noise. At 130M and 350M, the ranking is clear (5 of 6 pairwise gaps beyond noise). Fira is closest to full-rank and SLTrain is furthest. Surprisingly, ReLoRA is also close to full-rank (second at 130M, tied first at 350M), despite having the worst perplexity. *Takeaway: Each method converges to its own basin, and a loss barrier separates every pair of methods.*

5.2.1 Activation metrics.

Do low-rank methods compute similar representations to full-rank? Figure 4 shows activation deviation from full-rank, averaged over layers (Row 1) and at the last layer (Row 2). Averaged over layers, ReLoRA deviates least at 130M and 350M, and CoLA and Fira deviate most at 350M. Fira is the only method whose deviation drops at the last layer at every scale (e.g., $2.81 \rightarrow 2.49$ at 350M). The last decoder block plus final RMSNorm reduces the deviation, and Fira benefits from this more than any other method. **No method is closest to full-rank on every metric** (130M and 350M, beyond noise): GaLore and Fira are closest in L_2 distance and cosine similarity, ReLoRA is closest in CKA, and CoLA is furthest on all three. Per layer, the deviation grows with depth and over training (Row 3; all methods in Appendix E.3). *Takeaway: Despite similar perplexity, low-rank methods learn different representations from full-rank, and the gap grows with training and depth.*

5.2.2 Rank and spectral metrics

Do low-rank methods match full-rank in spectrum and rank? Figure 5 tracks rank and spectral metrics at 350M for the trained weights W (Row 1), the updates ΔW between consecutive checkpoints (Row 2), and the singular values of the total update per projection (Row 3). We materialize SLTrain and ReLoRA into full matrices and exclude CoLA, whose nonlinearity has no equivalent

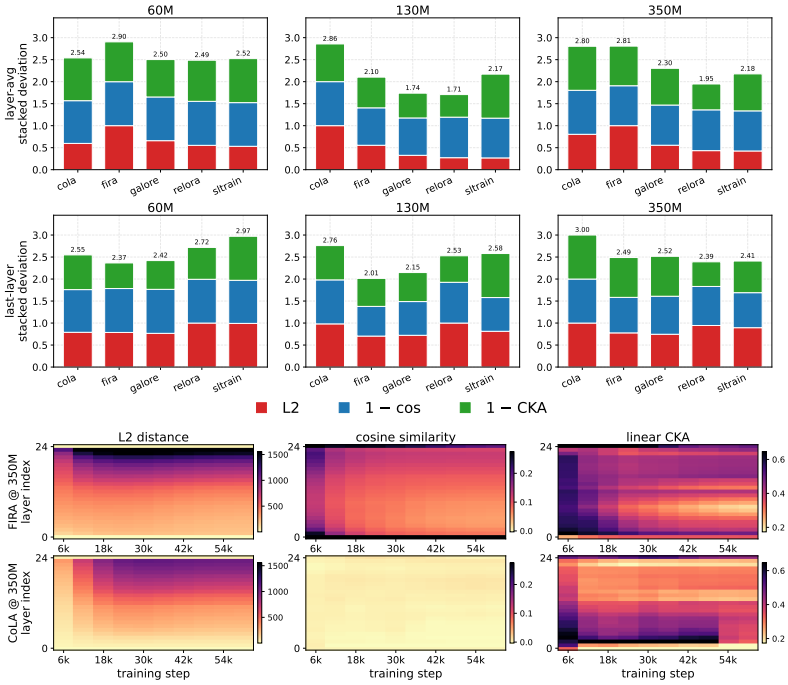


Figure 4: Activation deviation with full-rank baseline.

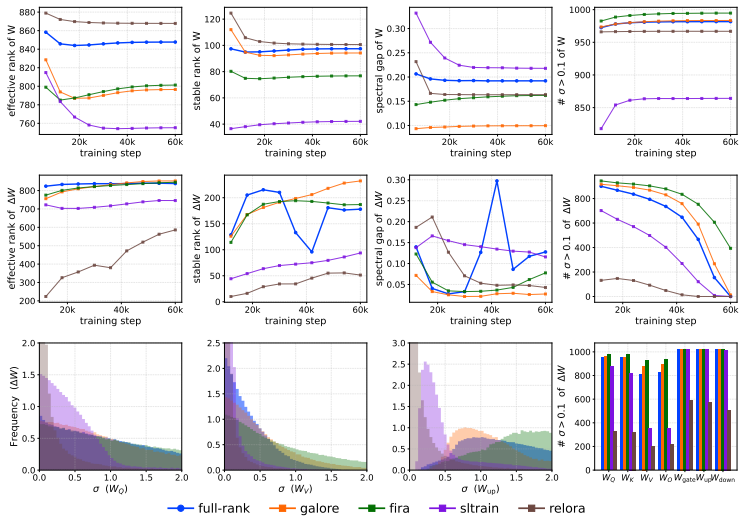


Figure 5: Rank and spectral metrics at 350M.

weight matrix. **Weights** W**.** On stable rank, the order is ReLoRA $\approx$ full-rank $\approx$ GaLore $>$ Fira $\gg$ SLTrain. SLTrain is lowest on every rank metric and has the largest spectral gap. GaLore matches full-rank on rank but has half its spectral gap (0.10 vs. 0.19). **Updates** ΔW**.** Here the order changes: GaLore $\approx$ Fira $\approx$ full-rank $>$ SLTrain $>$ ReLoRA. GaLore and Fira make near full-rank updates throughout training. SLTrain and ReLoRA make much lower-rank updates, concentrated at small singular values. Similar patterns hold at 60M and 130M (Appendix E.2). *Takeaway: Even with similar perplexity, low-rank methods follow different paths in rank and spectrum, throughout the training. GaLore and Fira come closest to full-rank; ReLoRA and SLTrain deviate the most.*

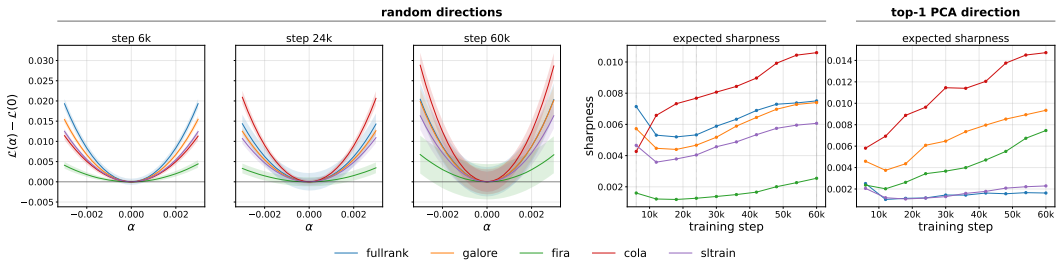
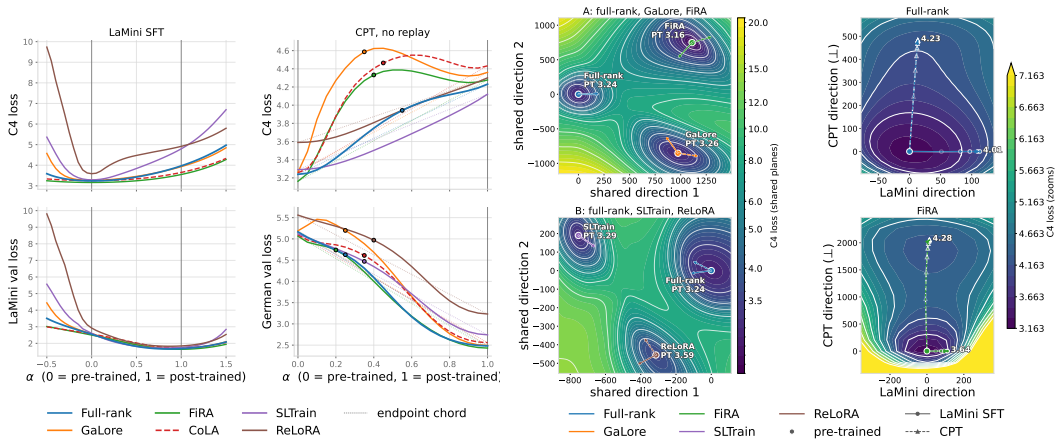


Figure 6: 1-D loss landscape at 350M along random directions (top) and the top-1 PCA direction (bottom). Left: centered loss $\mathcal{L}(\alpha) - \mathcal{L}(0)$ at four checkpoints. Right: sharpness over training.



(a) 1-D interpolation

(b) 2-D landscapes

Figure 7: Basin shape and post-training at 130M. **(a)** C4 loss (top) and target loss (bottom) along the path from pre-trained ($\alpha = 0$) to post-trained ($\alpha = 1$) checkpoints, for LaMini SFT (left) and CPT (right). Dots mark the peak above the endpoint chord (dotted). **(b)** 2-D C4 loss landscapes around the pre-trained checkpoints, with the LaMini and CPT paths.

5.2.3 Intrinsic geometry via 1-D loss landscape

Do low-rank methods converge to basins as sharp as full-rank? Figure 6 shows the loss around each checkpoint and its sharpness over training at 350M; ReLoRA and other scales are in Appendix E.1. Along random directions, **sharpness first decreases and then increases over training** for every method except CoLA. CoLA instead sharpens steadily throughout. ReLoRA is by far the sharpest at every scale; Fira and SLTrain are flatter than full-rank, while others are within noise. Along the top-1 PCA direction, full-rank and SLTrain are the flattest at every scale, while GaLore, CoLA, and ReLoRA are consistently sharper; Fira moves from flattest at 60M to the sharp group at 350M. Full-rank is more anisotropic than the low-rank methods at 60M, but this gap closes by 350M. *Takeaway: Basin sharpness depends on the direction. Full-rank and SLTrain are flattest along the top PCA direction, while Fira is by far the flattest along random directions.*

5.3 BASIN explains post-training behavior

Based on the observations in Section 5.1, we now use BASIN to find answers to two questions: **(Q1)** Why do SLTrain gain less and forget more than the other methods during LaMini SFT? **(Q2)** Why does Fira forget less than full-rank during SFT, but more during CPT? We analyze both cases with the Barriers lens.

SLTrain and ReLoRA sit in narrow basins. If the pre-trained basin is sharp, even a small SFT step climbs a steep wall in pre-training loss, which shows up as forgetting; if it is flat, the model can adapt with little change in pre-training loss. Figure 7a interpolates from each pre-trained checkpoint ($\alpha = 0$)

to its LaMini SFT checkpoint ($\alpha = 1$). Target loss drops smoothly for every method, but C4 loss does not. ReLoRA crosses a barrier, with C4 loss jumping right after $\alpha = 0$, and SLTrain climbs a steep wall, with C4 loss rising faster than any other method's. Both sit in narrow basins (Figure 7b), while the other methods adapt along a flatter direction and their C4 loss changes much less. **The sharpest basins along the task direction forget the most.**

CPT pushes Fira over a barrier. Figure 7a (right) interpolates from each pre-trained checkpoint to its CPT checkpoint. German loss drops smoothly for every method except GaLore, but C4 loss takes two kinds of path. For full-rank, SLTrain, and ReLoRA, C4 loss stays near the chord, so they leave their basin without crossing a high barrier. For Fira, GaLore, and CoLA, C4 loss peaks above the chord mid-path, so they must climb a barrier to reach the German solution. The 2-D view (Figure 7b, right) shows why for Fira: along the CPT direction it leaves the basin but for LaMiNi SFT it doesn't. **A basin that is flat for one task can still be steep for another.** More landscapes and CPT with replay are in Appendix G.

Takeaway: The properties of the solution a pre-trained model converges to matter. Post-training behavior depends on how sharp or flat the basin is along the direction of the new task.

5.4 Case studies: from diagnosis to design

BASIN can also guide how to improve low-rank methods (Figure 8). Fira deviates from full-rank in a positive direction and SLTrain in a negative one; in both cases, we aim to bring the method closer to full-rank so that its post-training behavior matches. We run both studies at 130M on a single seed. **Case study 1: Fira. Diagnosis.** Fira has a tail-heavy spectrum, with more weak singular directions than full-rank (e.g., W_{up} in Figure 5). GaLore, which is Fira without its off-subspace residual, has a head-heavy spectrum, and full-rank lies in between. **Fix.** We scale the residual by 0.5. **Result.** This narrows the gap to full-rank by 34.6% in directions with $\sigma > 0.1$, 16.8% in L_2 distance, 11.3% in sharpness, and 5% in barrier height. Perplexity, zero-shot accuracy, and LaMini gain stay about the same, while LaMini forgetting moves 37.6% closer to full-rank. **Case study 2: SLTrain. Diagnosis.** SLTrain has a high barrier to full-rank (Figure 3) from the start and a large spectral gap (Figure 5), so we hypothesize training restricted to a few directions. Since $W = sBA + S$, the fix should let BA, S, or both explore more directions. **Fix.** Since initialization governs what low-rank training can reach (Shivagunde et al., 2024), we initialize B and A as scaled orthonormal (QR) factors, so sBA starts with a flat spectrum instead of $B_0 = 0$. Following Du et al. (2025), we express S in an orthonormal DCT basis, $S \mapsto T_m^\top S T_n$, a fixed rotation so that each coefficient spans the whole matrix instead of a single entry. The parameter count is unchanged. **Result** The fix narrows the gap to full-rank by 47.7% in stable rank, 31.7% in $\sigma > 0.1$, 90.4% in basin sharpness and 9.8% in barrier height, and the mean gap across the top five singular directions falls from 0.088 to 0.077 against full-rank's 0.081. The behavioural metrics move as well: zero-shot accuracy is 71.8% closer, LaMini gain 13.0% closer and LaMini forgetting 63.3% closer, at a cost of 0.042 nats of validation loss.

6 Conclusion

Perplexity is not enough to judge low-rank pre-training: methods with similar perplexity behave differently after fine-tuning and continual pre-training. BASIN shows that each method converges to its own basin, with a distinct spectrum and representations, and the basin's shape along the task direction tracks how much the model learns and forgets during post-training. In two case studies, BASIN also guided fixes that moved low-rank closer to full-rank training. We hope BASIN helps future low-rank methods be judged by how they behave after pre-training, not only by their loss.

7 Limitations

We compute our metrics on LLaMA-style models below 1B parameters; generalization to billion-scale models, other architectures, datasets, and optimizers is not explored. We adopt two perturbation-direction choices for the 1-D loss landscape, but the most appropriate normalization for cross-method comparison in LLMs remains open. BASIN points to many possible interventions, and we explore only two, at a single scale.

AI USE STATEMENT

We used generative AI tools to edit and shorten the text of this paper, fix grammar, check consistency between the text and figures, and format LaTeX (tables, captions, and citation commands). We used AI tools for plotting figures. We used AI tools to help us implement logic and apply verification to it. We did not use generative AI tools to generate research ideas or make research decisions. We used AI tools to schedule and run experiments while hyperparameter tuning or making bash scripts to run a huge volume of experiments. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies the training behavior of low-rank pre-training methods and does not involve human subjects or personal data. All models are trained on public datasets (C4, German C4, and LaMini) and evaluated on public benchmarks through lm-evaluation-harness. Our models are small (at most 350M parameters) and are not intended for deployment. By making low-rank pre-training easier to evaluate and improve, this work may lower the memory cost of training language models; we do not foresee specific negative societal impacts beyond those of language modeling research in general.

REPRODUCIBILITY STATEMENT

Section 4 and Appendix B give the full pre-training, fine-tuning, and continual pre-training configurations, including learning rates, schedules, batch sizes, and sequence lengths (Tables 2–3). All metric definitions are in Appendices C.1, C.2, and F. All results are averaged over three seeds (except case studies), and we report orderings only where they hold beyond seed noise. All experiments fit on a single RTX 6000 Ada 48GB GPU. We will release our BASIN framework and code publicly for pre-training, post-training, metric computation, and evaluation upon acceptance.

REFERENCES

Maksym Andriushchenko, Francesco Croce, Maximilian Müller, Matthias Hein, and Nicolas Flammarion. A modern look at the relationship between sharpness and generalization. *arXiv preprint arXiv:2302.07011*, 2023.

Huanran Chen, Yinpeng Dong, Zeming Wei, Yao Huang, Yichi Zhang, Hang Su, and Jun Zhu. Unveiling the basin-like loss landscape in large language models. *arXiv preprint arXiv:2505.17646*, 2025.

Xi Chen, Kaituo Feng, Changsheng Li, Xunhao Lai, Xiangyu Yue, Ye Yuan, and Guoren Wang. Fira: Can we achieve full-rank training of llms under low-rank constraint? *ArXiv*, abs/2410.01623, 2024. URL https://api.semanticscholar.org/CorpusID:273026172.

Zhekai Du, Yinjie Min, Jingjing Li, Ke Lu, Changliang Zou, Liuhua Peng, Tingjin Chu, and Mingming Gong. Loca: Location-aware cosine adaptation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, volume 2025, pp. 86234–86274, 2025.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International conference on machine learning*, pp. 3259–3269. PMLR, 2020.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Andi Han, Jiaxiang Li, Wei Huang, Mingyi Hong, Akiko Takeda, Pratik Jawanpuria, and Bamdev Mishra. SLTrain: a sparse plus low-rank approach for parameter and memory efficient pretraining. *arXiv preprint arXiv:2406.02214*, 2024.

Yongchang Hao, Yanshuai Cao, and Lili Mou. Flora: Low-rank adapters are secretly gradient compressors. *arXiv preprint arXiv:2402.03293*, 2024.

Weihao Huang, Zhenyu Zhang, Yushun Zhang, Zhi-Quan Luo, Ruoyu Sun, and Zhangyang Wang. Galore-mini: Low rank gradient learning with fewer learning rates. In *NeurIPS 2024 Workshop on Fine-Tuning in Modern Machine Learning: Principles and Scalability*, 2024.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *arXiv preprint arXiv:2403.08763*, 2024.

AJAY KUMAR JAISWAL, Lu Yin, Zhenyu Zhang, Shiwei Liu, Jiawei Zhao, Yuandong Tian, and Zhangyang Wang. From galore to welore: How low-rank weights non-uniformly emerge from low-rank gradients. URL https://arxiv.org/html/2407.11239v1#:~:text=In%20this%20section%2C%20we%20investigate,GaLore%20in%20different%20compression%20ratios.

Simran Kaur, Jeremy Cohen, and Zachary Chase Lipton. On the maximum hessian eigenvalue and generalization. In Javier Antorán, Arno Blaas, Fan Feng, Sahra Ghalebikesabi, Ian Mason, Melanie F. Pradier, David Rohde, Francisco J. R. Ruiz, and Aaron Schein (eds.), *Proceedings on "I Can't Believe It's Not Better! - Understanding Deep Learning Through Empirical Falsification" at NeurIPS 2022 Workshops*, volume 187 of *Proceedings of Machine Learning Research*, pp. 51–65. PMLR, 03 Dec 2023. URL https://proceedings.mlr.press/v187/kaur23a.html.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning (ICML)*, 2019. URL https://arxiv.org/abs/1905.00414.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. URL https://arxiv.org/abs/1712.09913.

Jiaxi Li, Lu Yin, Li Shen, Jinjin Xu, Liwu Xu, Tianjin Huang, Wenwu Wang, Shiwei Liu, and Xilu Wang. Lost: Low-rank and sparse pre-training for large language models. *arXiv preprint arXiv:2508.02668*, 2025.

Tao Li, Zhengbao He, Yujun Li, Yasheng Wang, Lifeng Shang, and Xiaolin Huang. Flat-lora: Low-rank adaptation over a flat loss landscape. *arXiv preprint arXiv:2409.14396*, 2024.

Vladislav Lialin, Sherin Muckatira, Namrata Shivagunde, and Anna Rumshisky. ReLoRA: High-Rank Training Through Low-Rank Updates. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=DLJznSp6X3.

Hong Liu, Sang Michael Xie, Zhiyuan Li, and Tengyu Ma. Same pre-training loss, better downstream: Implicit bias matters for language models. In *International Conference on Machine Learning*, pp. 22188–22214. PMLR, 2023.

Xu-Hui Liu, Yali Du, Jun Wang, and Yang Yu. On the optimization landscape of low rank adaptation methods for large language models. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Ziyue Liu, Ruijie Zhang, Zhengyang Wang, Mingsong Yan, Zi Yang, Paul D Hovland, Bogdan Nicolae, Franck Cappello, Sui Tang, and Zheng Zhang. Cola: Compute-efficient pre-training of llms via low-rank activation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4627–4645, 2025b.

Sebastian Loeschcke, Mads Toftrup, Michael J Kastoryano, Serge Belongie, and Vésteinn Snæbjarnarson. Loqt: Low-rank adapters for quantized pretraining. *Advances in Neural Information Processing Systems*, 37:115282–115308, 2024.

Roy Miles, Pradyumna Reddy, Ismail Elezi, and Jiankang Deng. Velora: Memory efficient training using rank-1 sub-token projections. *Advances in Neural Information Processing Systems*, 37: 42292–42310, 2024.

Aashiq Muhamed, Oscar Li, David Woodruff, Mona Diab, and Virginia Smith. Grass: Compute efficient low-memory llm training with structured sparse gradients. *arXiv preprint arXiv:2406.17660*, 2024.

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

Namrata Shivagunde, Mayank Kulkarni, Giannis Karamanolakis, Jack G. M. FitzGerald, Yannick Versley, Saleh Soltan, Volkan Cevher, Jianhua Lu, and Anna Rumshisky. Approximations may be all you need: Towards pre-training llms with low-rank decomposition and optimizers. 2024. URL https://www.amazon.science/publications/approximations-may-be-all-you-need-towards-pre-training-llms-with-low-rank-decompo

DiJia Su, Andrew Gu, Jane Xu, Yuan Tian, and Jiawei Zhao. Galore 2: Large-scale llm pre-training by gradient low-rank projection. *ArXiv*, abs/2504.20437, 2025. URL https://api.semanticscholar.org/CorpusID:278171328.

Kaiyue Wen, Zhiyuan Li, Jason Wang, David Hall, Percy Liang, and Tengyu Ma. Understanding warmup-stable-decay learning rates: A river valley loss landscape perspective. *arXiv preprint arXiv:2410.05192*, 2024.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Fikri Aji. Lamini-lm: A diverse herd of distilled models from large-scale instructions. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 944–964, 2024.

Jinqi Xiao, Shen Sang, Tiancheng Zhi, Jing Liu, Qing Yan, Linjie Luo, and Bo Yuan. Coap: Memory-efficient training with correlation-aware gradient projection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30116–30126, 2025.

Zhenyu Zhang, Ajay Jaiswal, Lu Yin, Shiwei Liu, Jiawei Zhao, Yuandong Tian, and Zhangyang Wang. Q-galore: Quantized galore with int4 projection and layer-adaptive low-rank gradients. *arXiv preprint arXiv:2407.08296*, 2024.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. GaLore: Memory-Efficient LLM Training by Gradient Low-Rank Projection. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=hYHsrKDiX7.

Kaiye Zhou, Shucheng Wang, and Jun Xu. Switchlora: Switched low-rank adaptation can learn full-rank information. *arXiv preprint arXiv:2406.06564*, 2024.

Luca Zhou, Bo Zhao, Rose Yu, and Emanuele Rodolà. Demystifying mergeability: Interpretable properties to predict model merging success. *arXiv preprint arXiv:2601.22285*, 2026.

A FLOW-CHART FOR BASIN

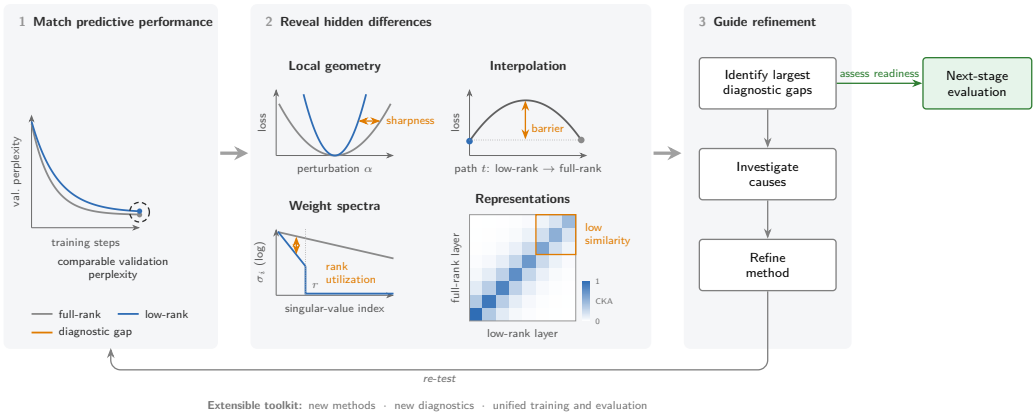


Figure 8: Flowchart on how to use BASIN

B ADDITIONAL DETAILS ON EXPERIMENTAL-SETUP

B.1 TASK-SFT

We fine-tune each method's pre-trained checkpoint on five downstream task and compare post-SFT performance against its zero-shot performance. For each method, we use the final pre-training checkpoint (e.g. step 10,000 on C4 for 60M) as the SFT initialization, so the only quantity that varies across methods is the pre-trained solution itself; the fine-tuning procedure is held fixed (see Shared-recipe protocol in Table 2). Low-rank checkpoints are materialized to their equivalent dense weights and fine-tuned as full-rank models, except CoLA, which is fine-tuned in its native parameterization. We fine-tune on the five zero-shot benchmarks that provide a training split: HellaSwag, PIQA, SWAG, ARC-Easy, and LogiQA2.

All tasks are multiple-choice and are scored by `lm-evaluation-harness` via the log-likelihood the model assigns to each candidate continuation. To keep the training signal aligned with this metric, we fine-tune as a causal language model with a completion-only objective and minimize the cross-entropy on the continuation tokens only, masking the prompt and padding. Because the goal is to isolate the effect of the pre-trained initialization, the fine-tuning recipe must not be re-tuned per method-otherwise differences in post-SFT accuracy could reflect differences in hyperparameter search rather than in the solutions themselves. We therefore tune the recipe once, on the full-rank model, independently per task: we sweep the learning rate over $\{1\times10^{-5}, 5\times10^{-5}, 1\times10^{-4}, 3\times10^{-4}, 1\times10^{-3}, 3\times10^{-3}\}$ and select by held-out validation performance with early stopping. The winning configuration fixes both the learning rate and the training-step budget for that task. All six methods are then fine-tuned with this identical, frozen recipe. We use AdamW (β_1=0.9, β_2=0.95, weight decay 0.01) with a linear warmup over the first 10% of steps followed by cosine decay, gradient clipping at 1.0, an effective batch size of 32, and `bfloat16` precision. The maximum sequence length is 512 for the longer-context tasks (HellaSwag, SWAG, LogiQA2) and 256 otherwise. Each (method, task) configuration is run with three random seeds, and we report the mean and standard deviation. The held-out set used for learning-rate selection and early stopping is chosen from each task's training split with a fixed seed, so the official validation/test split on which we report final accuracy is never seen during fine-tuning or model selection. Each fine-tuned checkpoint is evaluated with `lm-evaluation-harness` on its task's official evaluation split, using the same accuracy (acc/acc_norm) metrics and the same harness configuration as the zero-shot evaluation.

B.2 PART 2.1 LAMINI-SFT

Dataset. To study model plasticity to acquire new skill of instruction following, we fine-tune our pre-trained models on LaMini (Wu et al., 2024), a 2.58M-example instruction dataset drawn from seven sources (`original_flan`, `original_p3`, `generated_flan`, `generated_p3`, `self_instruct_with_topic`, `self_instruct_without_topic`, and `alpaca`). These seven sources collapse into three behavioral regimes: templated NLP tasks (long prompt → short answer; ≈77% of the data), open-ended generation (short prompt → long answer; ≈21%), and classic imperative instruction-following (≈2%). We profile the dataset in figures 9 and 10 and find that ≈95% of examples fit within a sequence

Setting	Value
Pre-training scale	60M, 130M, 350M (LLaMA-style, C4)
SFT initialization	final pre-training checkpoint
Methods	Full-rank, GaLore, Fira, CoLA, SLTrain, ReLoRA
Low-rank handling	materialized to dense (CoLA native)
Objective	causal LM, completion-only (prompt masked)
Optimizer	AdamW (β_1=0.9, β_2=0.95)
Weight decay	0.01
LR schedule	10% linear warmup $\rightarrow$ cosine decay
Gradient clipping	1.0
Precision	bfloat16
Effective batch size	32
Max sequence length	512 (long) / 256 (short)
Learning rate	tuned on full-rank per task over $\{1\times10^{-5}, 5\times10^{-5}, 1\times10^{-4}, 3\times10^{-4}, 1\times10^{-3}, 3\times10^{-3}\}$, then frozen
Step budget	set by full-rank early stopping, then frozen
Model selection	held-out val (carved from train), early stop
Seeds	3 (report mean $\pm$ std)

Table 2: Supervised fine-tuning configuration for the task-specific adaptation study. The recipe is tuned once on the full-rank model per task and applied identically to all six methods.

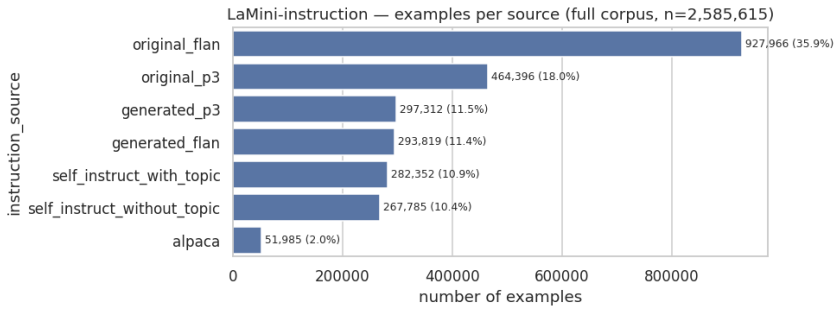


Figure 9: LaMini profile

length of 256 tokens, which both matches our pre-training context and lets us analyze adaptation *per category* of instruction. **Validation split:** LaMini ships without a validation split, so we construct a frozen, leakage-proof split by content-hash partitioning. From the validation pool we sample an equal-N set of 1,500 examples per category (10,500 total) **Training set:** The seven sources are highly imbalanced—`alpaca` contributes only ≈52k examples while `original_flan` contributes ≈928k—so we cap each category to build a balanced training set of ≈40k examples per source (≈266k total). All examples are tokenized once at a maximum length of 256 tokens with the minimal template `"Instruction: {instruction} \nResponse:"`, using completion-only masking so the loss is computed on the response and EOS token only.

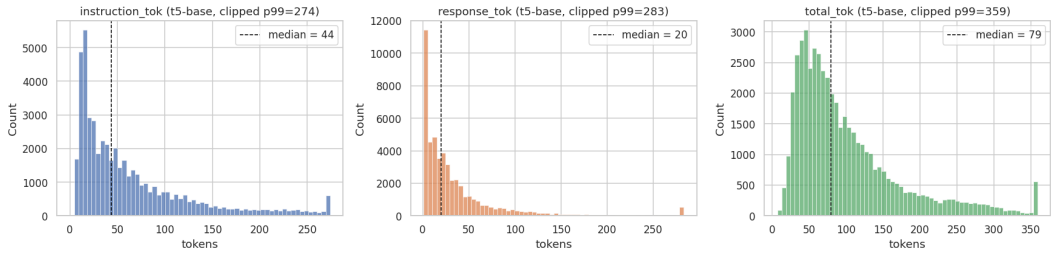


Figure 10: LaMini sequence length profile

Training protocol. We warm-start from the final pre-training checkpoint of each method (full-rank, GaLore, Fira, CoLA, SLTrain, ReLoRA) in weights-only mode: parameters are loaded while optimizer state, scheduler, and step counters reset. SFT reuses the pre-training trainer and method integrations, so each checkpoint is fine-tuned with same recipe. We tune the learning rate and use 1×10^{-3} at 60M and 3×10^{-4} at 130M and 350M. All runs use AdamW with a cosine schedule (1.5% warm up steps, min-LR ratio 0.1), no weight decay, gradient clipping 1.0, effective batch size 128, and three seeds.

B.3 CPT

Task-SFT and LaMini-SFT both adapt a model *within* its pre-training distribution: the target text is English web-like prose the model has already seen billions of tokens of. Continual pre-training (CPT) on a new language tests the opposite regime—acquiring a distribution the pre-trained model has essentially no coverage of—and it cleanly separates *plasticity* (how fast German loss falls) from *retention* (how much English is destroyed along the way).

Dataset. We continue pre-training on German C4 (`allenai/c4`, config `de`), for 20,000 update steps at 130M—matching the original pre-training budget for that scale (≈2.3B tokens at an effective batch of 512 × 256 tokens). We run CPT at 130M only: it is the scale at which the methods are already separated on both axes while remaining affordable across two protocols and three seeds. **Two protocols.** *No replay* trains on 100% German. *25% replay* mixes in the original pre-training distribution—the standard defense against catastrophic forgetting—and is the direct test of whether the method-level forgetting differences survive when forgetting is actively resisted. Mixing is per-*document* via `interleave_datasets` with probabilities 0.75/0.25 over (`c4_de`, `c4_en`), so every batch is a mixture rather than alternating batches.

Tokenizer. Following standard continual pre-training practice (Ibrahim et al., 2024), we keep the `t5-base` tokenizer fixed across pre-training, SFT and CPT. The German runs ask an English-trained vocabulary to encode a new language, and every method pays exactly the same cost. Measured on 1,500 C4 validation documents per language (Table 4), German costs $1.23\times$ more tokens per character and $1.49\times$ more per word than English; against a size-matched German tokenizer (GBERT), `t5-base` needs 20% more tokens for the same German text and the two vocabularies share $\approx 30\%$ of their types. One consequence is worth stating: because German documents are $1.95\times$ more tokens, the $75/25$ *document* mixture delivers a realized $85.4/14.6$ token split. Vocabulary extension and embedding re-initialization would reduce this overhead but add a second, method-dependent variable, and are beyond the scope of our work. Since tokenizer, corpus and budget are identical across methods, the handicap is a constant offset that leaves the between-method comparison—our object of study—unaffected; we read German loss as a relative measure across methods, not as an estimate of attainable German quality.

Training protocol. We take each method's final pre-trained checkpoint and simply continue pre-training with the *same* method: the low-rank methods are carried over unchanged, so CPT is a continuation of the original training procedure on a new corpus rather than a new adaptation scheme. Optimizer state is not restored—each run starts from a fresh optimizer, which is the natural choice for a new training phase. The learning rate is tuned per method and lies in $[1\times10^{-3},\ 5\times10^{-3}]$ (full-rank, SLTrain and ReLoRA 1×10^{-3}; CoLA 3×10^{-3}; GaLore and Fira 5×10^{-3}), re-warmed over the first 1% of the budget and then cosine-decayed. Remaining hyperparameters are shared: effective batch size 512 (micro-batch 128, gradient accumulation 4), maximum sequence length 256, gradient clipping 1.0, and `bfloat16` precision and three seeds.

Metrics. We evaluate on a held-out German split and on the same held-out English C4 subset used throughout the paper. We report three quantities: *German val loss* (the target), *gain* (reduction in German val loss relative to the pre-CPT model), and *forgetting* (increase in held-out English C4 loss relative to the pre-CPT model).

C More details on metrics

We provide more details on the metrics in this section.

C.1 Activation-based metrics.

To measure how closely a low-rank checkpoint's internal representations match those of the full-rank baseline, we compare their hidden states on identical inputs. Using the same fixed evaluation subset and seed as in section 3.4, we run both the full-rank baseline f_A and the low-rank target f_B and extract the hidden states at every layer $\ell \in \{0, 1, \ldots, L\}$, where $\ell = 0$ denotes the embedding output. The resulting per-layer activation matrices are

$$H_A^{(\ell)} \in \mathbb{R}^{N\times d}, \qquad H_B^{(\ell)} \in \mathbb{R}^{N\times d},$$

Setting	Value
Scale	130M only
CPT initialization	each method's final pre-training checkpoint
Methods	Full-rank, GaLore, Fira, CoLA, SLTrain, ReLoRA
Target corpus	German C4 (`allenai/c4`, `de`),
Protocols	no replay (100% de)
	25% replay (de/en $= 0.75/0.25$ per document)
Realized token split (replay)	85.4% German / 14.6% English
Objective	causal LM (standard pre-training loss)
Optimizer	each method's own pre-training optimizer
Learning rate	lr sweep starting from each method's own pre-training peak, re-warmed
	full-rank 1×10^{-3}, GaLore 5×10^{-3}, Fira 5×10^{-3},
	CoLA 3×10^{-3}, SLTrain 1×10^{-3}, ReLoRA 1×10^{-3}
LR schedule	1% re-warmup $\rightarrow$ cosine decay
Gradient clipping	1.0
Precision	bfloat16
Effective batch size	512 (micro-batch 128, accumulation 4)
Max sequence length	256
Training steps	20,000 ($\approx$2.3B tokens)
Seeds	3

Table 3: Continual pre-training configuration.

	English	German	DE/EN
tokens per character	0.2354	0.2904	1.23×
tokens per word	1.399	2.089	1.49×
characters per word	5.943	7.194	1.21×
mean tokens per document	502.1	979.6	1.95×
`<unk>` per 1k tokens	1.50	2.91	1.94×

Table 4: The `t5-base` tokenizer on English vs. German C4 (1,500 validation documents per language).

respectively, where N is the total number of token positions in a sequence and d is the hidden dimension. Let $h_{A,i}^{(\ell)}$ and $h_{B,i}^{(\ell)}$ denote the i-th row of $H_A^{(\ell)}$ and $H_B^{(\ell)}$ respectively, i.e. the hidden activation vectors at position i. We then report the following per-layer metrics.

- **Activation L2 distance:** the mean per-position Euclidean distance between the two activation matrices,

$$d_{\text{L2}}^{(\ell)} \;=\; \frac{1}{N}\sum_{i=1}^{N}\left\|h_{A,i}^{(\ell)} - h_{B,i}^{(\ell)}\right\|_2. \tag{1}$$

  A value of 0 means the two checkpoints produce identical activations on the input, and larger values indicate that the low-rank model departs further from the baseline in the absolute feature space.

- **Activation cosine similarity:** the mean per-position directional alignment between the two activation vectors,

$$\cos^{(\ell)} \;=\; \frac{1}{N}\sum_{i=1}^{N}\frac{\langle h_{A,i}^{(\ell)},\, h_{B,i}^{(\ell)}\rangle}{\|h_{A,i}^{(\ell)}\|_2\,\|h_{B,i}^{(\ell)}\|_2 + \varepsilon}. \tag{2}$$

  A value of 1 indicates that, at every position, the low-rank model points in the same direction in feature space as the baseline; values near 0 indicate orthogonal, unrelated features. Unlike the L2 distance, this is insensitive to activation magnitude and isolates the directional component of the mismatch.

- **Linear CKA:** following (Kornblith et al., 2019), we compute the linear Centered Kernel Alignment between the two full activation matrices as a representation-level similarity. Let $\tilde{H}_A^{(\ell)}$ and $\tilde{H}_B^{(\ell)}$ denote the column-centered versions of $H_A^{(\ell)}$ and $H_B^{(\ell)}$, and define the cross- and self-Gram matrices

$$G_{AB} \;=\; \big(\tilde{H}_B^{(\ell)}\big)^\top \tilde{H}_A^{(\ell)}, \qquad G_{AA} \;=\; \big(\tilde{H}_A^{(\ell)}\big)^\top \tilde{H}_A^{(\ell)}, \qquad G_{BB} \;=\; \big(\tilde{H}_B^{(\ell)}\big)^\top \tilde{H}_B^{(\ell)}.$$

Linear CKA is then

$$\mathrm{CKA}^{(\ell)} \;=\; \frac{\|G_{AB}\|_F^2}{\|G_{AA}\|_F \, \|G_{BB}\|_F}. \tag{3}$$

Linear CKA is invariant to orthogonal rotations and isotropic scaling of the hidden features, so it captures whether the two checkpoints encode the same representational subspace even when individual neurons have been permuted or re-scaled. A value of 1 indicates perfectly aligned subspaces and a value near 0 indicates unrelated ones.

C.2 Rank related metrics

Since each low-rank pre-training method imposes a different structural prior on the weight matrices it learns, we examine the singular value distribution of the trained checkpoints to characterize how that prior manifests across the training horizon. These metrics surface complementary aspects of the spectrum: how much of the matrix's energy concentrates in the leading components, how broadly the remaining mass is distributed, and how many directions actively contribute to the matrix's action. Following the probe family adopted by (Zhou et al., 2026) in their study of task-vector alignment, we restrict our analysis to the attention and MLP weight matrices, as these are the layers where the low-rank parametrization is enforced. For each two-dimensional weight $W \in \mathbb{R}^{m\times n}$, we obtain the ordered singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$ with $r = \min(m, n)$ and compute the following quantities.

- **Effective rank:** The exponential of the entropy of the normalized spectrum, treating singular values as a probability distribution:

$$\mathrm{EffRank}(W) \;=\; \exp\!\left(-\sum_{i=1}^{r} p_i \log p_i\right), \qquad p_i \;=\; \frac{\sigma_i}{\sum_j \sigma_j}. \tag{4}$$

  This quantity equals 1 when the spectrum collapses entirely onto a single direction and approaches r as the singular values flatten toward uniformity.

- **Stable rank:** A spectrum-based dimensionality measure defined as the ratio of the Frobenius norm to the operator norm,

$$\mathrm{StableRank}(W) \;=\; \frac{\|W\|_F^2}{\|W\|_2^2} \;=\; \frac{\sum_i \sigma_i^2}{\sigma_1^2}. \tag{5}$$

  It is upper-bounded by the algebraic rank but, unlike the latter, is insensitive to negligibly small singular values. A value of 1 corresponds to an essentially rank-one matrix, while a value approaching r reflects a near-uniform spectrum.

- **Spectral gap:** The relative separation between the leading two singular values, $\mathrm{SpectralGap}(W) = (\sigma_1 - \sigma_2)/\sigma_1$. A pronounced gap signals that the top direction stands out clearly from the rest of the spectrum; a vanishing gap means the first two directions carry comparable weight and are not robustly distinguishable.

- **Threshold rank:** A hard count of singular values whose magnitude exceeds a fixed cutoff $\tau = 0.1$, namely $\mathrm{TRank}_\tau(W) = \#\{\, i : \sigma_i > \tau \,\}$. This filters out tail components that are numerically nonzero but have negligible influence on the matrix's action, providing a noise-robust alternative to the strict rank.

D Downstream evaluation

We report downstream performance per task for 60M, 130M, 350M in Table 5, 6, and 7, respectively.

E More results

We report additional plots and results for all the metric families in this section.

E.1 1-D loss landscape

This section reports the full set of 1-D loss landscape plots complementing the 350M random-direction results and the top-1 PCA results in Figure 3 of the main paper. Figures 11–16 show random-direction landscapes at all three scales (60M, 130M, 350M), each presented twice — once excluding ReLoRA and once including it. Because ReLoRA's sharpness is one to two orders of magnitude larger than the other methods at every scale, including it compresses the y-axis and obscures the relative ordering among the remaining methods; we therefore provide both views. Figures 17–19 extend the top-k PCA-direction analysis to $k \in \{1, 5, 10, 20\}$ at each scale.

Task	LLaMA	GaLore	Fira	CoLA	SLTrain	ReLoRA
CS						
HellaSwag†	27.09 ± 0.21	27.83 ± 0.33	27.59 ± 0.46	27.64 ± 0.23	27.47 ± 0.39	27.13 ± 0.15
PIQA	59.36 ± 0.98	58.81 ± 0.58	59.27 ± 0.41	58.94 ± 0.30	59.14 ± 1.46	57.15 ± 0.78
COPA	55.33 ± 1.53	56.00 ± 3.00	57.33 ± 3.51	56.00 ± 1.73	54.67 ± 1.53	55.00 ± 1.00
SWAG†	36.51 ± 0.29	36.85 ± 0.14	**37.39 ± 0.16**	36.01 ± 0.93	36.48 ± 0.35	34.68 ± 0.32
WK						
OpenBookQA†	29.20 ± 0.72	30.00 ± 1.64	30.47 ± 0.31	29.93 ± 1.33	29.47 ± 1.01	30.00 ± 1.91
ARC-Easy	28.90 ± 0.69	29.87 ± 0.51	29.98 ± 0.58	29.15 ± 1.11	29.50 ± 0.23	28.96 ± 0.28
QA4MRE-2013†	23.71 ± 3.19	24.77 ± 2.24	23.12 ± 1.94	23.24 ± 0.70	25.12 ± 2.87	23.12 ± 3.53
RC						
ReCoRD‡	36.84 ± 0.46	37.27 ± 0.85	38.23 ± 1.81	39.61 ± 1.25	38.27 ± 1.75	39.56 ± 0.95
Gr.						
BLiMP	75.23 ± 0.62	73.89 ± 0.71	**75.39 ± 0.86**	73.45 ± 0.63	73.11 ± 0.16	71.89 ± 1.54
LR						
LogiQA†	26.68 ± 0.99	28.21 ± 0.54	27.04 ± 1.60	25.91 ± 0.18	26.32 ± 1.23	25.70 ± 0.62
LogiQA2†	26.44 ± 0.86	25.34 ± 0.89	26.10 ± 0.63	**27.18 ± 1.10**	25.30 ± 0.89	24.47 ± 0.55
Macro (z)	-0.10 ± 0.39	+0.29 ± 0.14	**+0.49 ± 0.39**	+0.10 ± 0.40	-0.07 ± 0.21	-0.71 ± 0.19

Table 5: **Zero-shot results at 60M**, mean ± std over three pre-training seeds (%). Metrics are lm-eval-harness defaults: † `acc_norm`, ‡ F1, `acc` otherwise. **Bold** marks the best method only where methods differ significantly (one-way ANOVA, permutation $p < 0.05$); rows with no bold are within seed noise. The macro row averages per-task z-scores for each (method, seed).

Task	LLaMA	GaLore	Fira	CoLA	SLTrain	ReLoRA
CS						
HellaSwag†	28.91 ± 0.42	29.22 ± 0.31	**29.53 ± 0.10**	28.93 ± 0.14	29.06 ± 0.40	27.19 ± 0.34
PIQA	60.88 ± 0.62	61.10 ± 0.71	**62.01 ± 0.36**	60.90 ± 0.17	61.43 ± 0.33	59.50 ± 0.17
COPA	61.00 ± 1.73	54.67 ± 3.06	**62.00 ± 4.58**	56.67 ± 1.53	58.00 ± 1.00	53.33 ± 0.58
SWAG†	39.15 ± 0.37	**39.83 ± 0.61**	39.83 ± 0.48	39.19 ± 0.29	38.84 ± 0.05	34.91 ± 0.19
WK						
OpenBookQA†	29.40 ± 0.35	30.40 ± 1.51	31.07 ± 1.55	30.13 ± 0.83	30.20 ± 0.72	29.67 ± 0.83
ARC-Easy	30.18 ± 0.18	30.47 ± 0.17	30.64 ± 0.11	**30.86 ± 0.61**	30.19 ± 0.72	29.29 ± 0.55
QA4MRE-2013†	22.54 ± 2.88	22.42 ± 1.08	23.83 ± 0.54	20.89 ± 1.81	23.71 ± 1.42	23.71 ± 1.77
RC						
ReCoRD‡	42.65 ± 1.41	40.42 ± 1.33	44.48 ± 3.01	45.57 ± 3.52	44.34 ± 1.70	41.09 ± 1.81
Gr.						
BLiMP	77.01 ± 0.93	76.39 ± 0.34	**77.67 ± 1.39**	77.22 ± 1.05	75.99 ± 1.86	74.20 ± 0.88
LR						
LogiQA†	27.19 ± 0.96	26.63 ± 1.19	26.93 ± 0.23	27.60 ± 1.39	27.50 ± 2.13	26.42 ± 0.41
LogiQA2†	26.99 ± 0.53	26.76 ± 0.75	27.10 ± 0.29	26.55 ± 0.24	26.53 ± 1.00	25.53 ± 0.49
Macro (z)	+0.10 ± 0.21	-0.02 ± 0.11	**+0.71 ± 0.17**	+0.17 ± 0.27	+0.18 ± 0.35	-1.14 ± 0.22

Table 6: **Zero-shot results at 130M**. Same setup as Table 5.

E.2 Rank related metrics

This section reports the rank and spectral metric trajectories at the 60M and 130M scales, complementing the 350M results shown in Figure 5 of the main paper. Both figures follow the same layout as Figure 5 *Row 1* tracks the four spectral metrics — effective rank, stable rank, spectral gap, and the count of singular values above 0.1 — on the trained weight W across training; *Row 2* tracks the same metrics on the consecutive-checkpoint update $\Delta W = W_{t+1} - W_t$; and *Row 3* shows the singular value distributions of ΔW per projection type (W_Q, W_V, W_{up}), with the rightmost panel reporting the per-projection count of singular values above 0.1. The same method ordering observed at 350M — GaLore ≈ Fira > SLTrain > CoLA > ReLoRA — holds

Task	LLaMA	GaLore	Fira	CoLA	SLTrain	ReLoRA
CS						
HellaSwag†	32.47 ± 0.67	32.05 ± 0.21	**34.84 ± 0.37**	31.98 ± 0.38	31.14 ± 0.10	28.03 ± 0.55
PIQA	63.15 ± 0.40	63.78 ± 0.87	**64.67 ± 0.60**	63.53 ± 1.02	62.89 ± 0.30	59.88 ± 0.22
COPA	64.33 ± 2.08	62.00 ± 2.00	62.00 ± 3.61	62.00 ± 4.36	60.67 ± 0.58	56.33 ± 3.51
SWAG†	42.37 ± 1.20	42.48 ± 0.59	**43.74 ± 1.22**	42.18 ± 0.59	41.24 ± 0.50	36.90 ± 0.98
WK						
OpenBookQA†	30.40 ± 1.11	33.13 ± 0.76	**34.53 ± 1.63**	33.47 ± 1.62	31.13 ± 0.92	30.73 ± 1.62
ARC-Easy	**31.62 ± 0.29**	31.07 ± 0.32	31.36 ± 0.58	31.50 ± 0.97	31.17 ± 0.51	29.15 ± 0.43
QA4MRE-2013†	22.54 ± 0.93	22.65 ± 3.04	**24.77 ± 1.13**	24.18 ± 1.59	23.59 ± 1.61	18.66 ± 2.44
RC						
ReCoRD‡	46.20 ± 2.31	42.34 ± 1.23	48.04 ± 0.94	**51.15 ± 4.84**	50.89 ± 4.28	42.14 ± 0.96
Gr.						
BLiMP	78.16 ± 2.39	77.73 ± 0.61	78.44 ± 1.99	77.22 ± 1.50	77.77 ± 2.12	75.49 ± 0.97
LR						
LogiQA†	25.96 ± 1.47	25.55 ± 2.43	25.96 ± 0.93	27.80 ± 1.41	27.50 ± 0.96	26.93 ± 1.13
LogiQA2†	25.93 ± 0.64	26.51 ± 0.88	27.08 ± 0.19	26.42 ± 0.35	26.65 ± 1.12	26.46 ± 0.90
Macro (z)	+0.03 ± 0.39	+0.02 ± 0.26	**+0.66 ± 0.14**	+0.38 ± 0.37	+0.11 ± 0.39	-1.20 ± 0.17

Table 7: **Zero-shot results at 350M**. Same setup as Table 5.

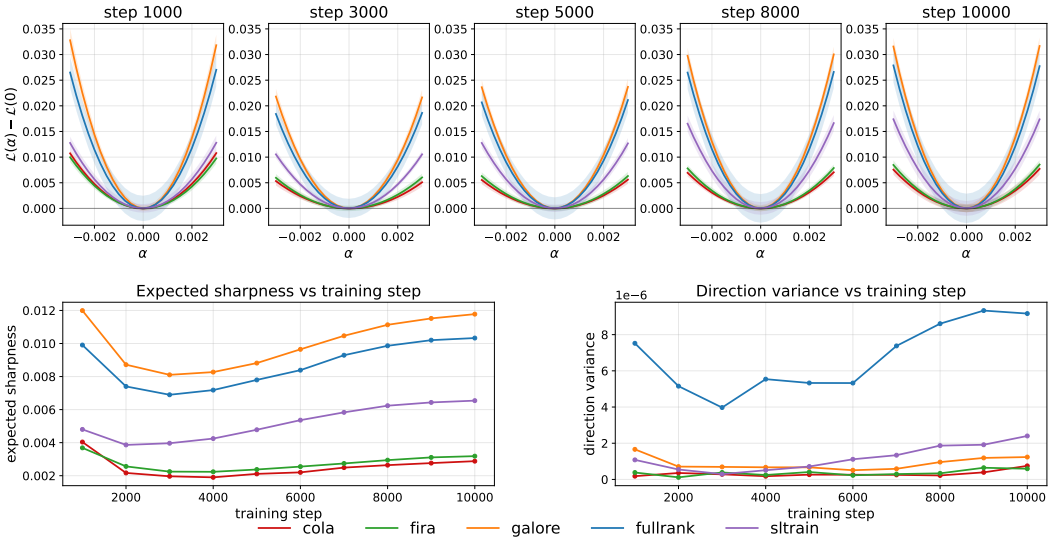


Figure 11: **1-D loss landscape at 60M parameters.** *Top:* centered loss profile $\mathcal{L}(\alpha) - \mathcal{L}(0)$ averaged over 100 random directions, at five training checkpoints (1k, 3k, 5k, 8k, 10k). *Bottom-left:* average sharpness with respect to training step. *Bottom-right:* average direction variance. **ReLoRA is omitted** from the plot as it makes it harder to view other methods. The plot including ReLoRA is given in 12

.

at both smaller scales: optimizer-based methods (GaLore, Fira) most faithfully reproduce full-rank's update geometry, while parameterization-based methods (CoLA, SLTrain) and ReLoRA systematically underpopulate the small-singular-value tail in the attention W_V/W_O blocks.

E.3 Activations

This section reports the full per-layer × per-training-step activation heatmaps for every method at every model scale, complementing the bar summaries and the Fira/CoLA-only heatmaps shown in Figure 4 of the main paper. All three figures share the same grid layout: rows index the five low-rank methods (GaLore, Fira, CoLA,

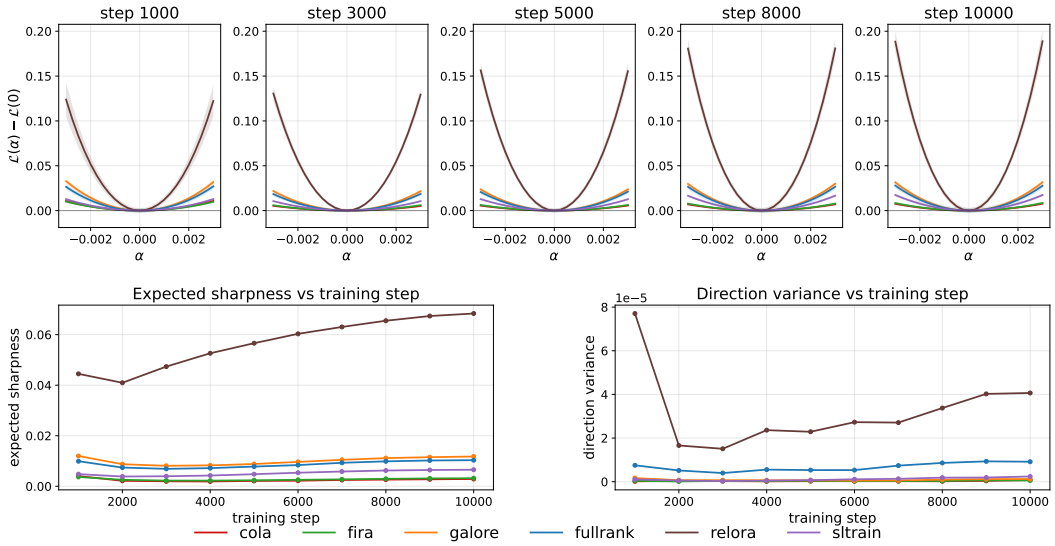


Figure 12: **1-D loss landscape at 60M parameters for All methods.** *Top:* centered loss profile $\mathcal{L}(\alpha) - \mathcal{L}(0)$ averaged over 100 random directions, at five training checkpoints (1k, 3k, 5k, 8k, 10k). *Bottom-left:* average expected sharpness with respect to training step. *Bottom-right:* average direction variance.

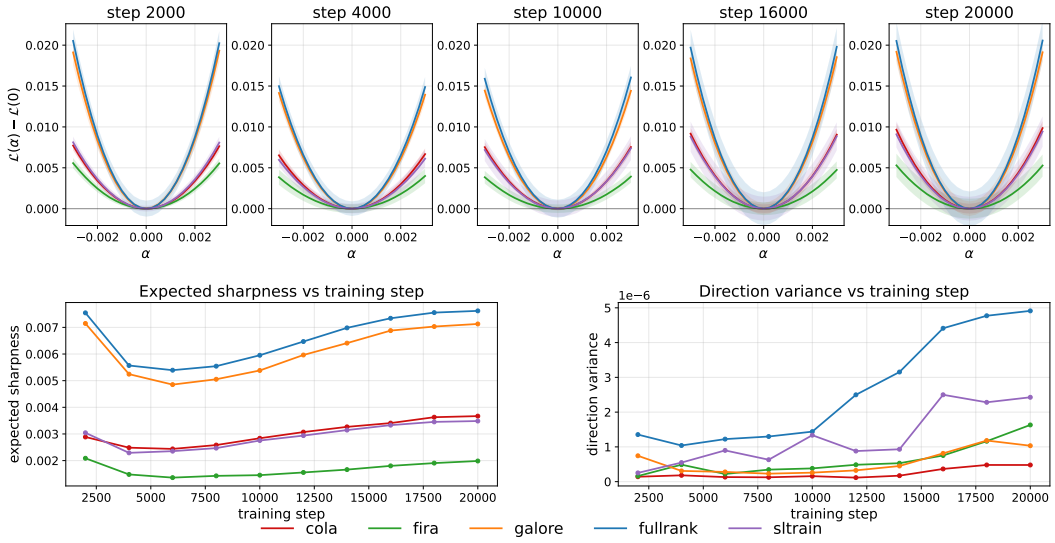


Figure 13: **1-D loss landscape at 130M parameters.** Same layout as Figure 11, with top-row checkpoints at $\{2\text{k}, 6\text{k}, 10\text{k}, 14\text{k}, 20\text{k}\}$.

SLTrain, ReLoRA) and columns index the three model scales (60M, 130M, 350M); within each panel, the x-axis is the training step and the y-axis is the decoder layer index, where layer 0 corresponds to the embedding output. Figure 22 reports the per-layer Activation L2 distance to the full-rank baseline (Eq. 1), Figure 23 reports the per-layer Linear CKA similarity (Eq. 3), and Figure 24 reports the per-layer Activation cosine similarity (Eq. 2). Across all three metrics, divergence from full-rank concentrates in the later decoder layers and grows with training, with CoLA showing the most severe directional mismatch — cosine similarity remains near zero across all layers and steps at 130M and 350M.

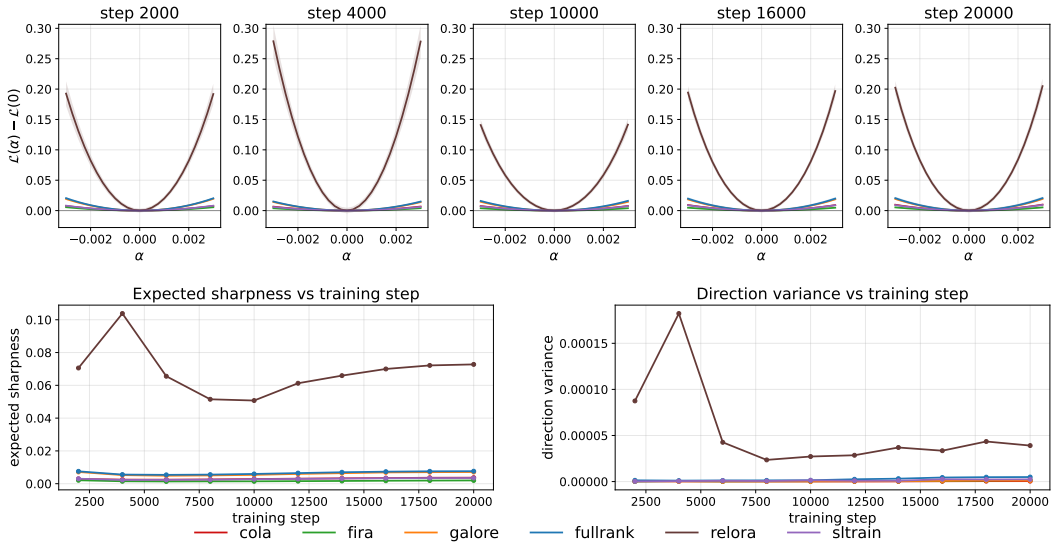


Figure 14: 1-D loss landscape at 130M parameters for **All methods**.

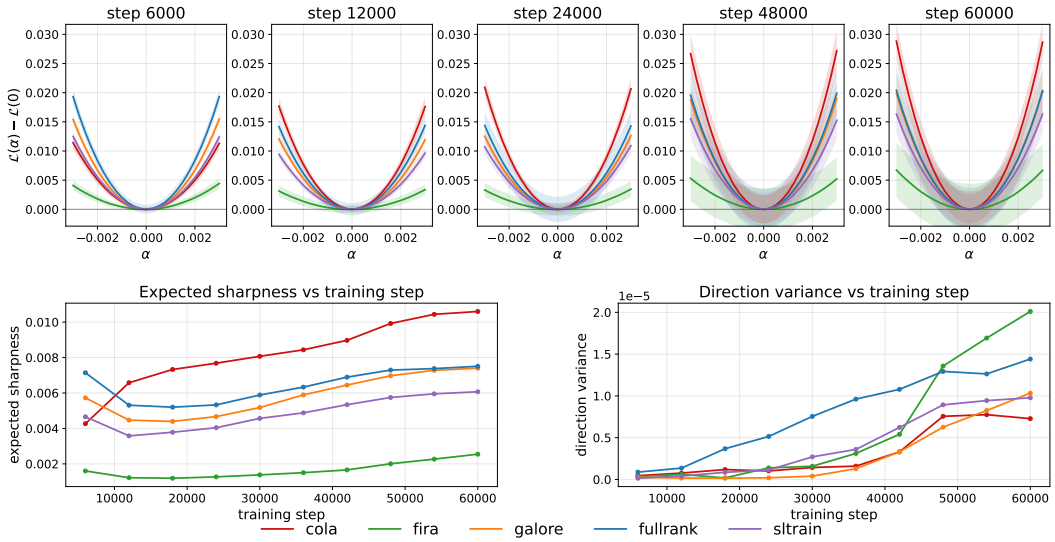


Figure 15: **1-D loss landscape at 350M parameters.** Same layout as 11.

F LOSS-LANDSCAPE DETAILS

To understand the local geometry of the loss landscape around each pre-training checkpoint, we perturb the model parameters along a single randomly chosen direction and record how the validation loss varies as we move along this direction. However, since a single direction only captures a d-dimensional slice and is not necessarily representative of the surrounding geometry, we repeat the process along 100 independently sampled random directions and record the mean and variance of the validation loss across these directions. Following (Chen et al., 2025) , we adapt the formula to our pre-training setting where the loss is the C4 (Raffel et al., 2020) validation loss (we pre-train our models on C4). For a single random direction δ: $\mathcal{L}(\alpha) = \mathcal{L}_{\text{val}}(\theta + \alpha\,\delta)\,, \qquad \delta \sim \mathcal{N}(0, I)$, where θ are the checkpoint parameters, $\alpha \in [\alpha_{\min}, \alpha_{\max}]$ is the perturbation magnitude swept on a uniform grid, and $\mathcal{L}_{\text{val}}$ is the validation loss. δ has the same dimensions as the parameter matrix it perturbs. We average $\mathcal{L}(\alpha)$ over 100 independently sampled directions $\delta_1, \ldots, \delta_D \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, I)$ where $D = 100$:

$$\bar{\mathcal{L}}(\alpha, \delta) \;=\; \frac{1}{D}\sum_{i=1}^{D} \mathcal{L}_{\text{val}}(\theta + \alpha\,\delta_i)\,, \qquad \sigma^2(\alpha, \delta) \;=\; \frac{1}{D}\sum_{i=1}^{D}\Big(\mathcal{L}_{\text{val}}(\theta + \alpha\,\delta_i) - \bar{\mathcal{L}}(\alpha)\Big)^2. \tag{6}$$

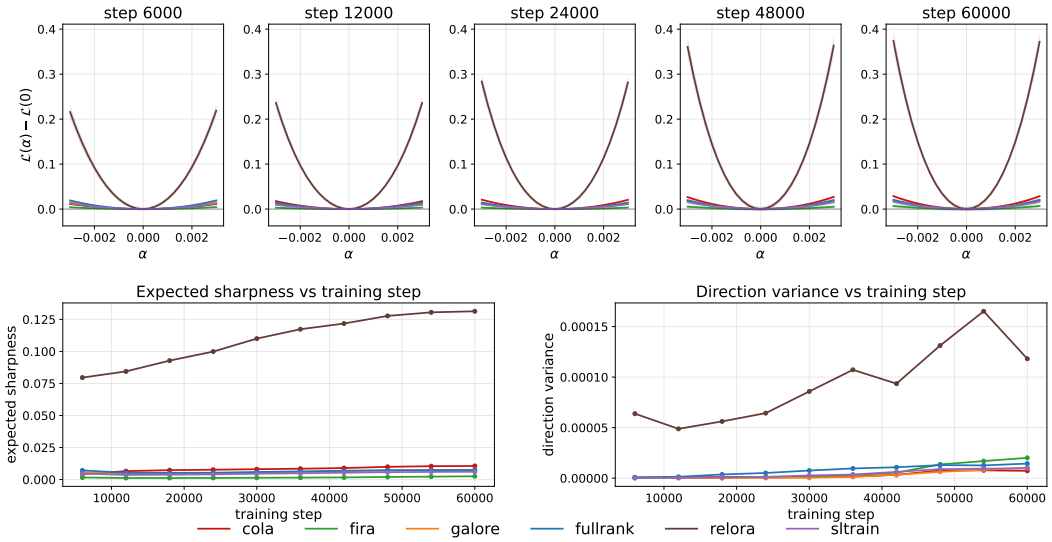


Figure 16: 1-D loss landscape at 350M parameters for **All methods**.

where $\bar{\mathcal{L}}(\alpha, \delta)$ denotes the mean of the validation loss across the D random directions at perturbation magnitude α, and $\sigma^2(\alpha, \delta)$ denotes its variance. For convenience, we drop δ in the notation.

Sharpness The 1-D loss landscape from the previous section yields a full curve $\bar{\mathcal{L}}(\alpha, \delta)$ per checkpoint, which is qualitative in nature. To compare methods quantitatively across checkpoints, model sizes, and training steps, we reduce each curve to a single scalar capturing how flat or sharp the local basin is around the checkpoint. Intuitively, the rate at which $\bar{\mathcal{L}}$ grows as we move away from the checkpoint ($\alpha = 0$) serves as a discrete proxy for local curvature: sharp curvature produces steep loss changes in the neighborhood, while flat curvature keeps losses close to the reference value at $\alpha = 0$. Concretely, for each direction δ_i, we evaluate the loss at symmetric perturbations $\pm\alpha_j$ on a grid of N offsets (where N is the number of positive offsets in the perturbation grid) and measure the elevation $\Delta(\delta_i, \alpha_j) = \mathcal{L}(\delta_i, \alpha_j) - \mathcal{L}(\delta_i, 0)$ relative to the checkpoint. Sharpness (S) is then the average of these elevations across all D directions and both signs of α:

$$S = \frac{1}{2ND} \sum_{i=0}^{D-1} \sum_{j=0}^{N-1} \Big[\Delta(\delta_i, +\alpha_j) + \Delta(\delta_i, -\alpha_j) \Big], \qquad \Delta(\delta, \alpha) = \mathcal{L}(\delta, \alpha) - \mathcal{L}(\delta, 0). \tag{7}$$

A larger S indicates a sharper basin; a smaller S indicates a flatter one.

Direction variance The variance band in the 1-D loss landscape plots is a qualitative metric, and direction variance is its quantitative counterpart. We compute it by averaging the across-direction variance $\sigma^2(\alpha, \delta)$ from equation 6 over the offsets N, similar to equation 7). See Appendix **??** for equations.

1-D loss landscape on top-K PCA directions. Random Gaussian directions probe most-case geometry around a checkpoint. We therefore complement them with a principled alternative: applying SVD to the weight matrix and perturbing along its top-k singular directions. Specifically, for every two-dimensional parameter tensor $W \in \mathbb{R}^{m \times n}$ in the checkpoint θ, we compute the SVD, $W = U\Sigma V^\top$ and form the top-k perturbation direction as $\delta^{(k)} = \sigma_k(u_k \otimes v_k)$, where u_k and v_k are the k-th left and right singular vectors and σ_k is the corresponding singular value. The resulting 1-D loss slice is $\mathcal{L}^{\text{pca}}(\alpha; k) = \mathcal{L}_{\text{val}}\Big(\theta + \alpha\, \delta^{(k)}\Big)$.

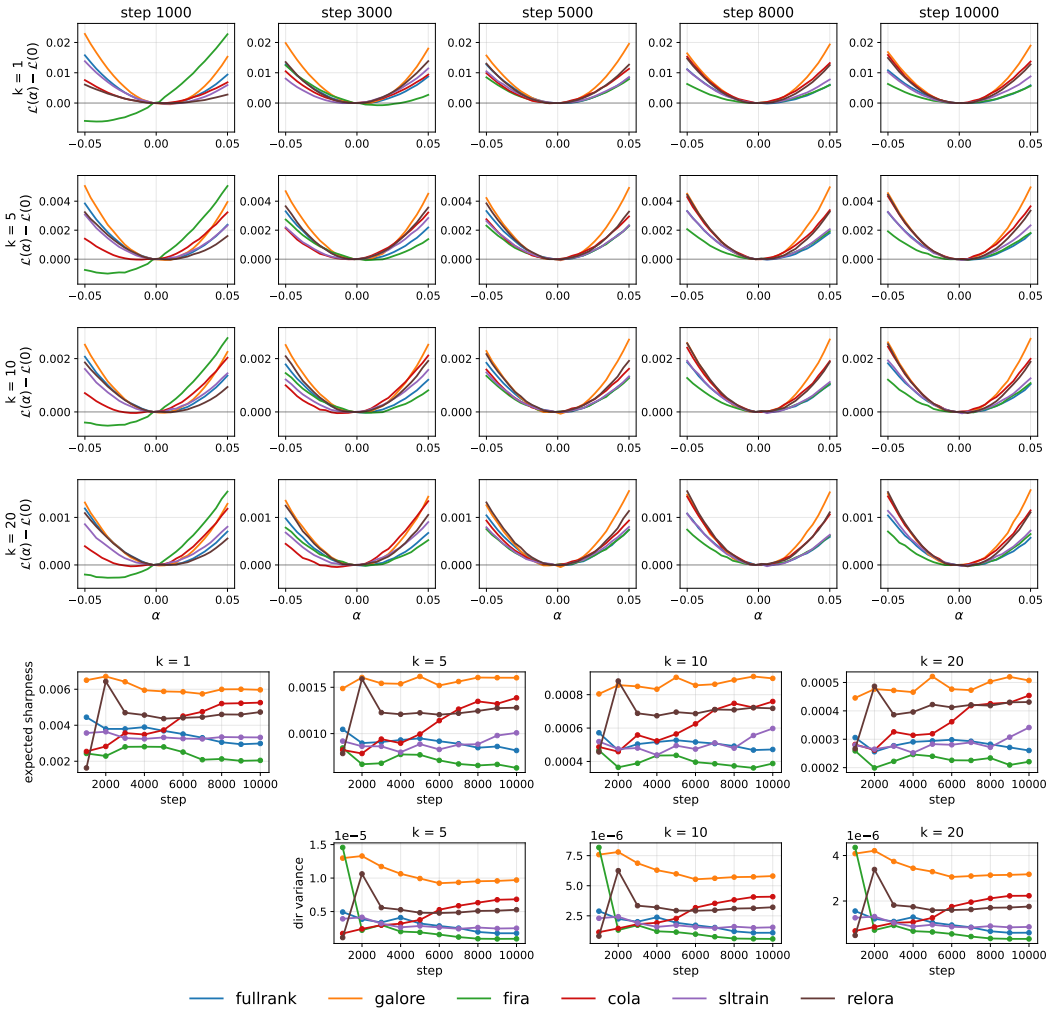


Figure 17: **1-D loss landscape along top-k PCA directions at 60M parameters** for $k \in \{1, 5, 10, 20\}$. *Top four rows:* centered loss profile at five training checkpoints. *Bottom:* expected sharpness (left column of summary panels) and across-component direction variance (right column) as a function of training step, per k.

G BASIN ANALYSIS OF POST-TRAINING: FULL RESULTS

This appendix extends Section 5.3. We show the 2-D C4 loss landscape for every method (Appendix G.2) and the 1-D interpolation for CPT with 25% replay (Appendix G.1). All results are at 130M. We measure a barrier as the height of the C4 loss curve above the straight chord between its two endpoints.

G.1 CPT WITH REPLAY

Figure 25 repeats the CPT interpolation with 25% C4 replay. We see the same pattern as without replay (Section 5.3). German loss drops smoothly for every method. For full-rank, SLTrain, and ReLoRA, C4 loss stays close to the chord, so CPT moves them without crossing a barrier. For Fira, GaLore, and CoLA, C4 loss rises above the chord and peaks mid-path, so they still cross a barrier on the way to the German solution. Replay lowers the C4 loss at the end of every path and shrinks the barriers for Fira and CoLA, but it does not remove them. GaLore's barrier stays the largest.

G.2 2-D LOSS LANDSCAPES FOR ALL METHODS

Figure 26 shows three things.

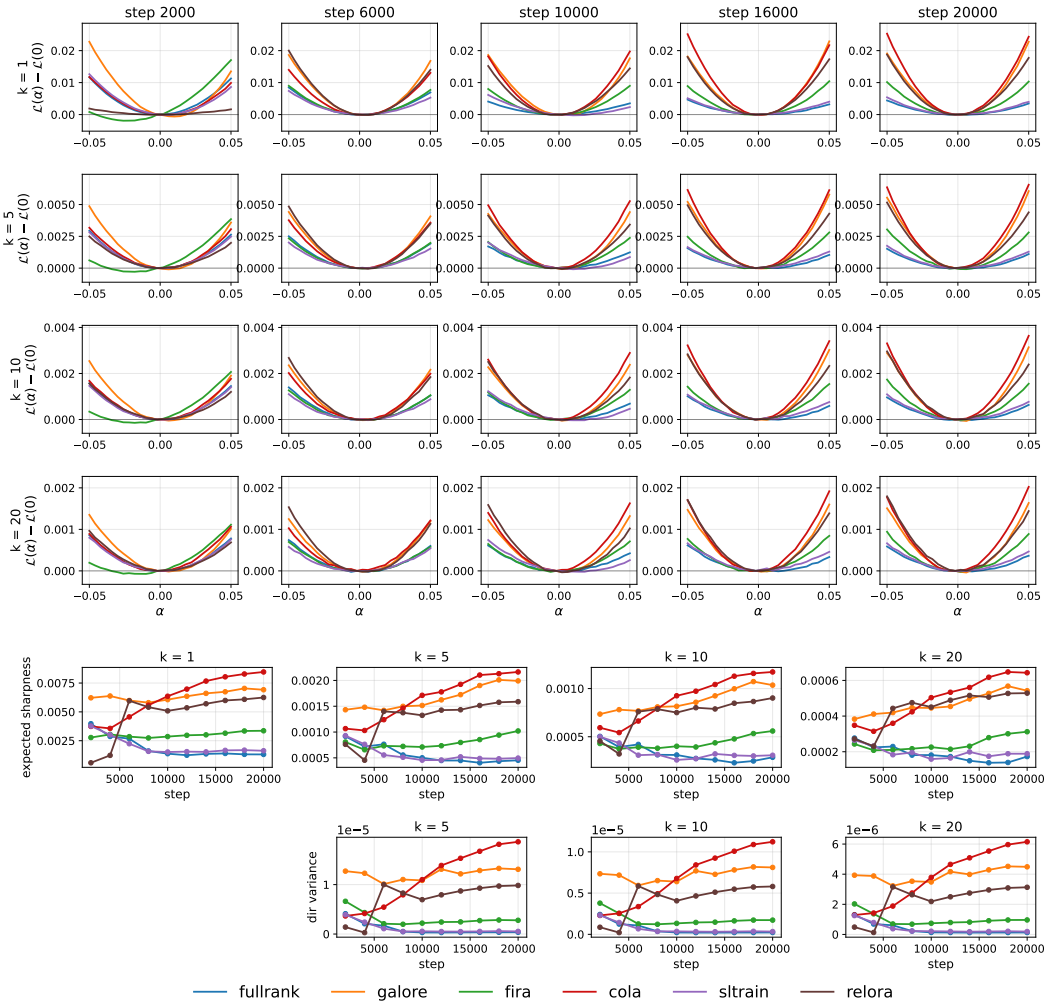


Figure 18: **1-D loss landscape along top-k PCA directions at 130M parameters** for $k \in \{1, 5, 10, 20\}$. Layout identical to Figure 17.

Each method converges to its own basin. On the shared planes (left column), every pre-trained checkpoint sits in a separate basin, with regions of high loss between them. This matches the large inter-method barriers in Section 5.2.

The basins have different shapes. Fira and full-rank sit in basins that are wide along the LaMini direction. SLTrain and ReLoRA sit in narrow basins with tightly packed contours. GaLore sits in a narrow channel with steep walls on both sides.

Post-training takes different paths. All methods move about the same distance for LaMini SFT, but the C4 loss they reach differs a lot. It rises by about 0.5 nats for Fira, 0.8 for full-rank, 0.9 for GaLore, 1.3 for ReLoRA, and 1.7 for SLTrain. Since the step length is similar, the difference comes from the basin shape. For CPT, the distance itself differs. Fira moves about four times farther than full-rank and leaves its basin, and GaLore moves about three times farther. SLTrain and ReLoRA move less than full-rank, which is why they forget less but also learn less German.

Takeaway: Low-rank methods converge to different solutions, and the landscape around each solution decides how the model behaves after post-training. How far post-training moves the model, and how steep the landscape is along that path, together set how much it forgets.

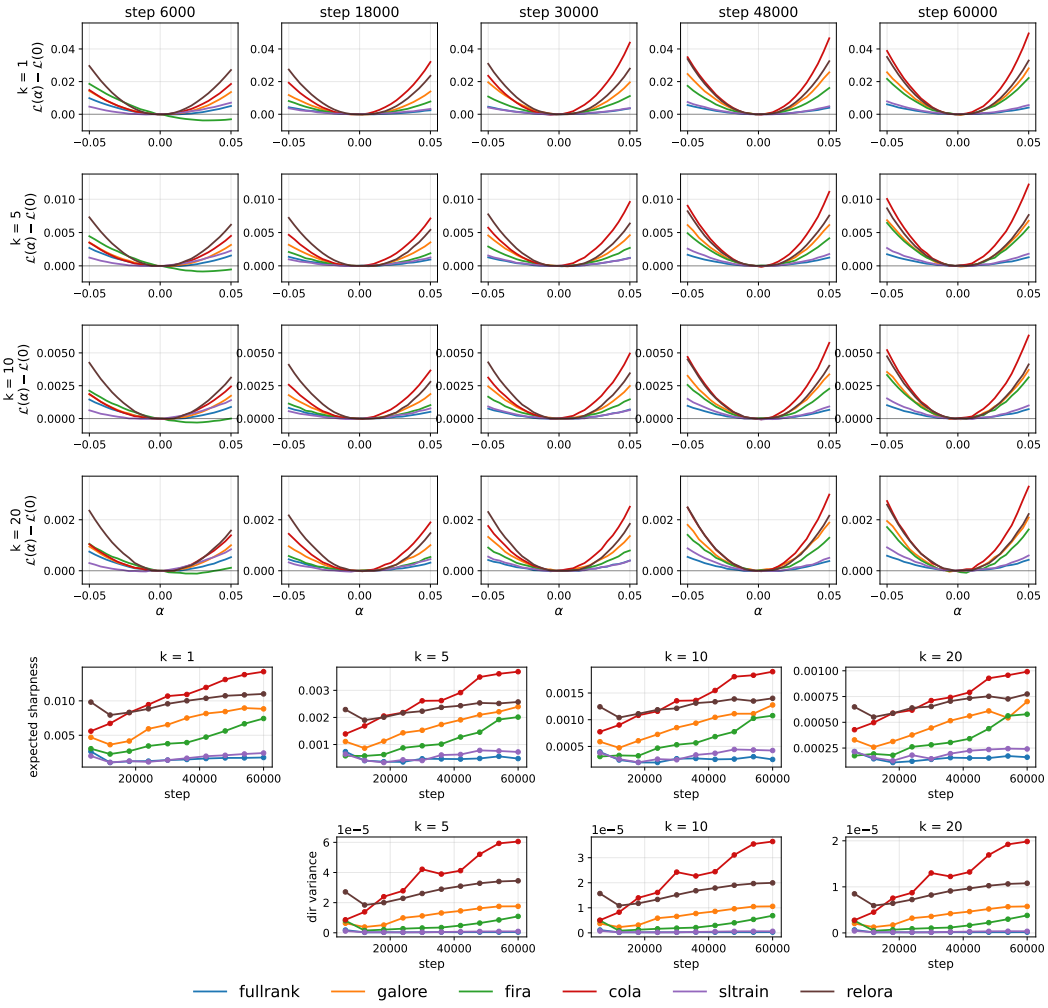


Figure 19: **1-D loss landscape along top-k PCA directions at 350M parameters** for $k \in \{1, 5, 10, 20\}$. Layout identical to Figure 17. CoLA's direction variance grows monotonically with training across all k, while other methods remain near zero, indicating that CoLA's sharpness is distributed anisotropically across its top-k subspace rather than concentrated in any single direction.

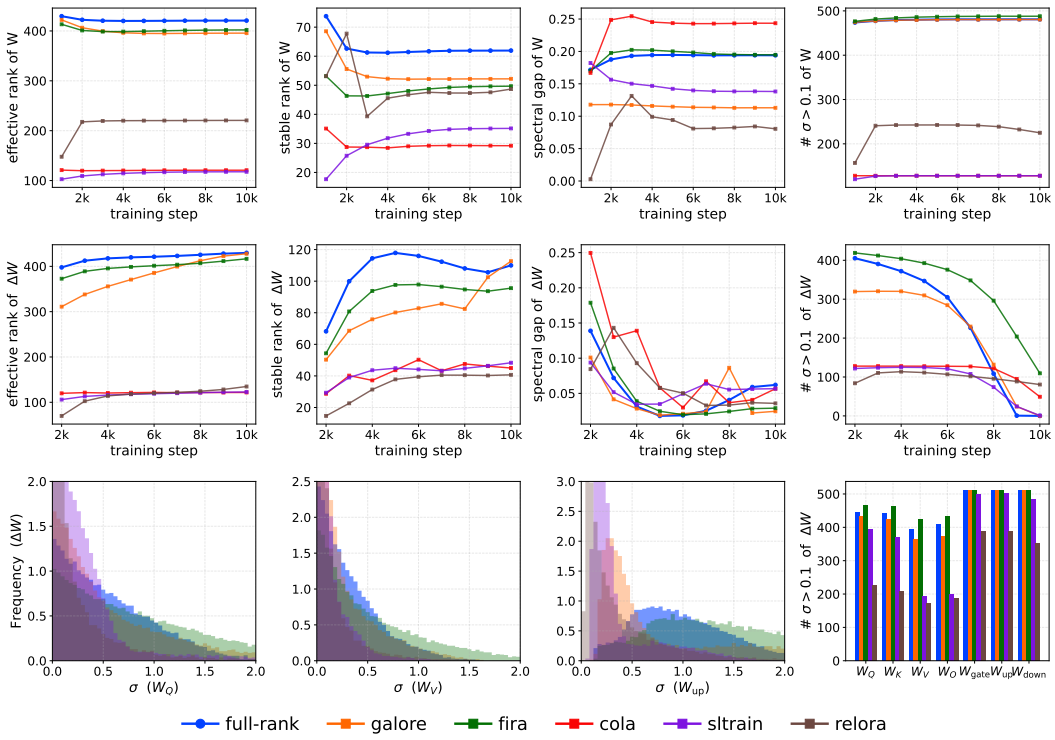


Figure 20: Rank and Spectral metrics for 60M.

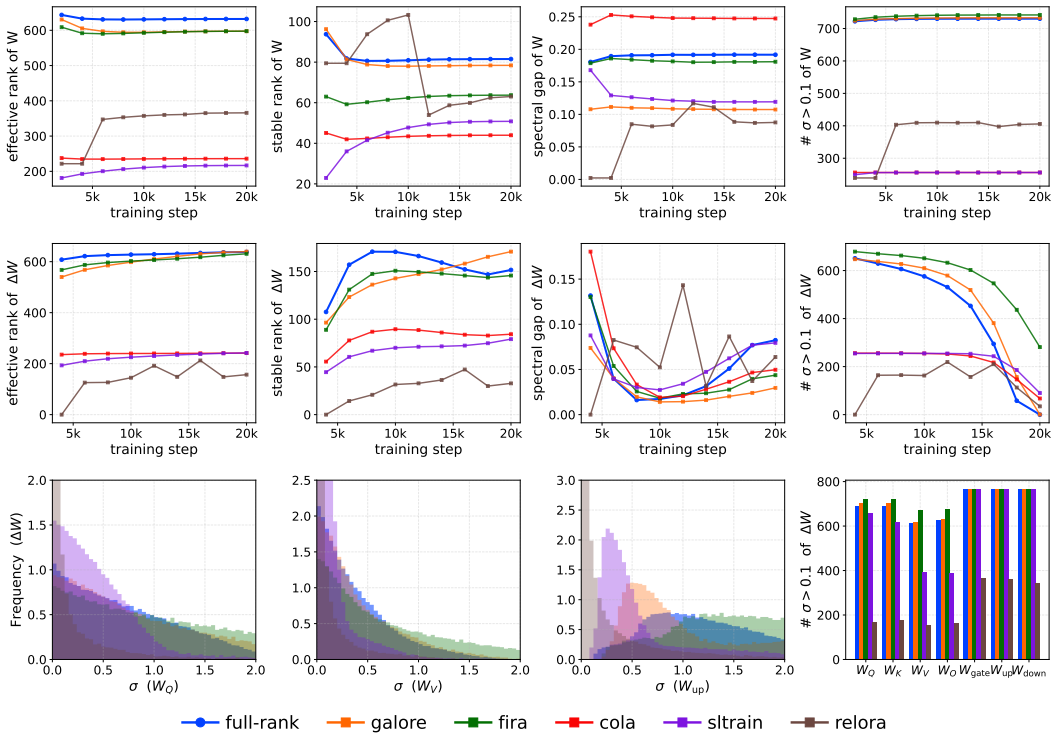


Figure 21: Rank and Spectral metrics for 130M.

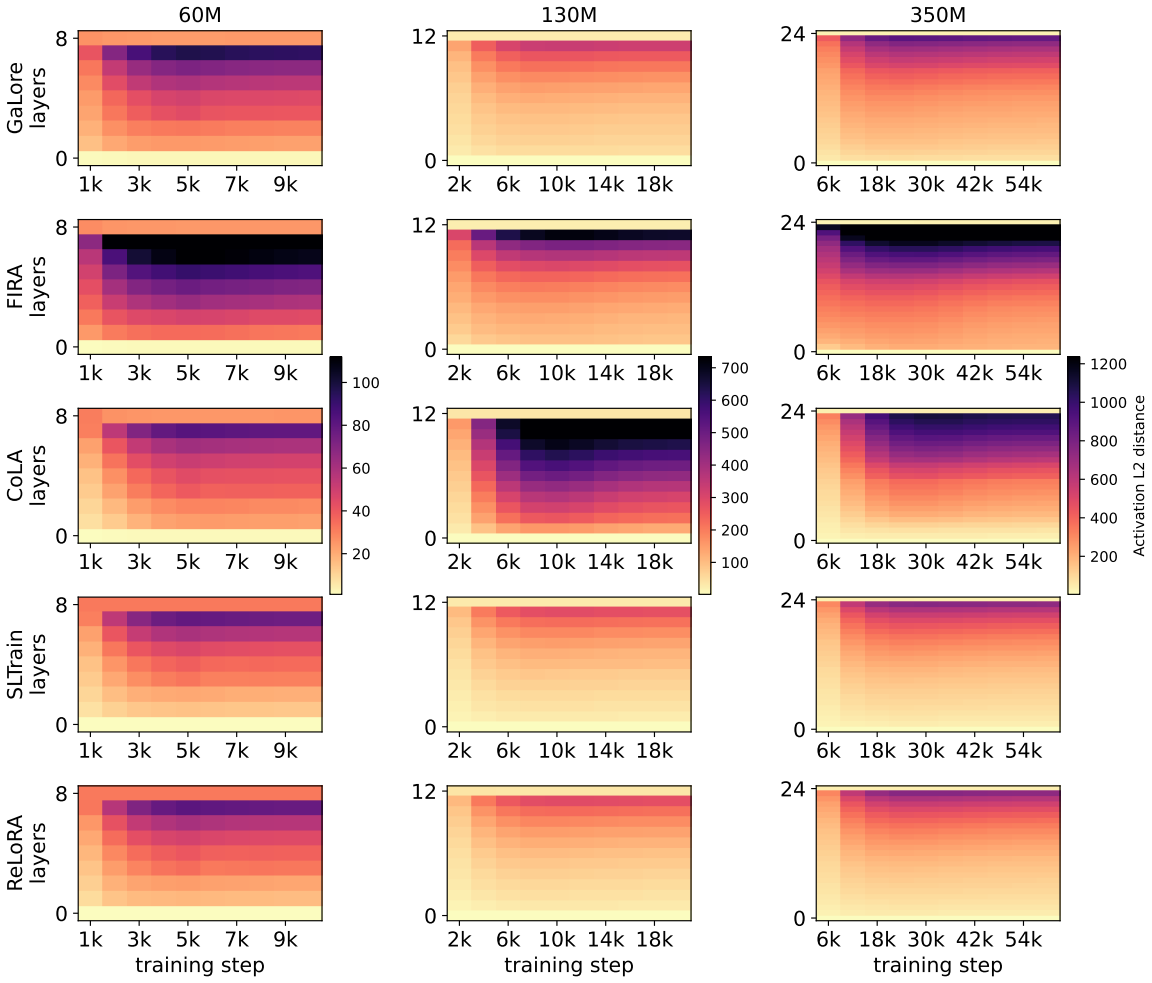


Figure 22: Activation L2 distance layer-wise

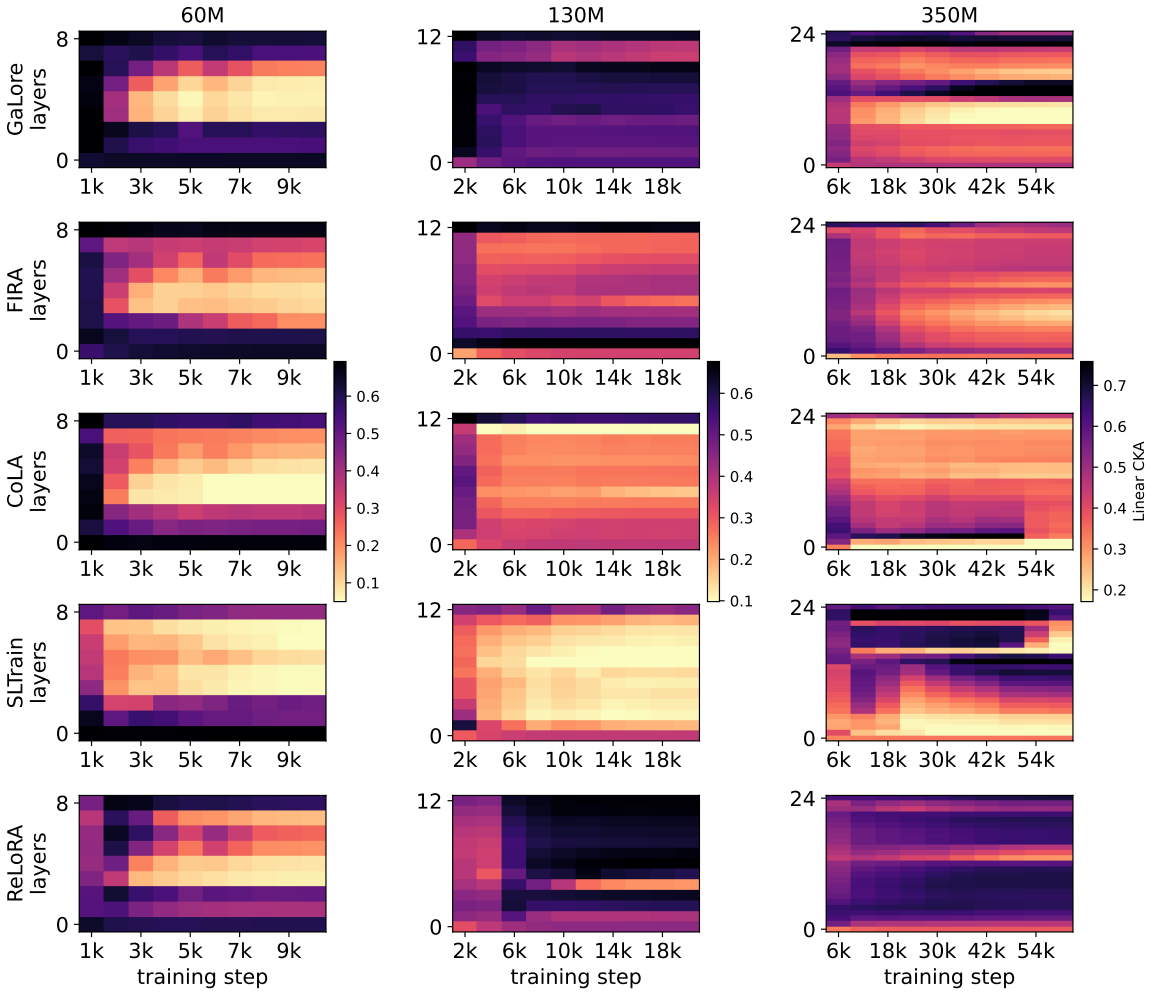


Figure 23: Activation linear CKA similarity

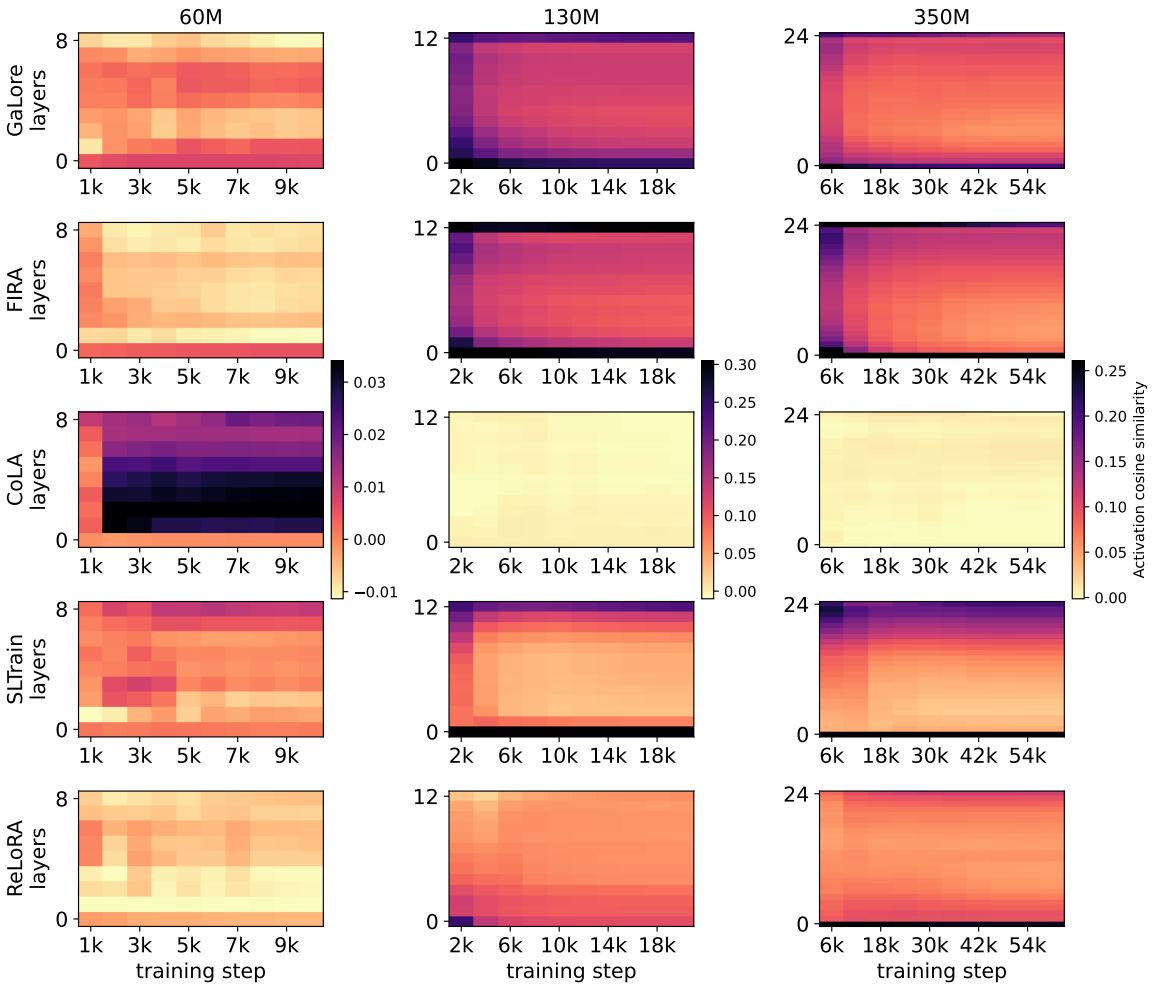


Figure 24: Activation cosine similarity layer-wise

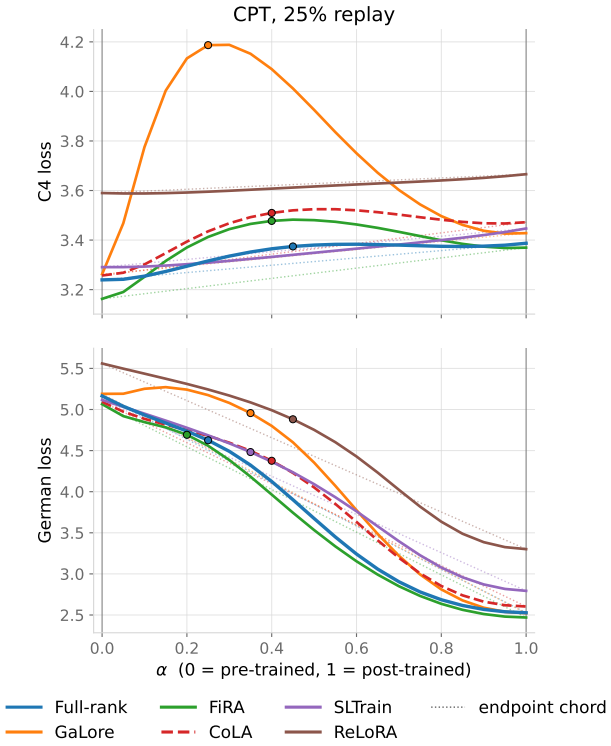


Figure 25: C4 loss (top) and German loss (bottom) along the straight path from each method's pre-trained checkpoint ($\alpha = 0$) to its CPT checkpoint with 25% C4 replay ($\alpha = 1$), at 130M. Dots mark the highest point of each curve above the endpoint chord (dotted).

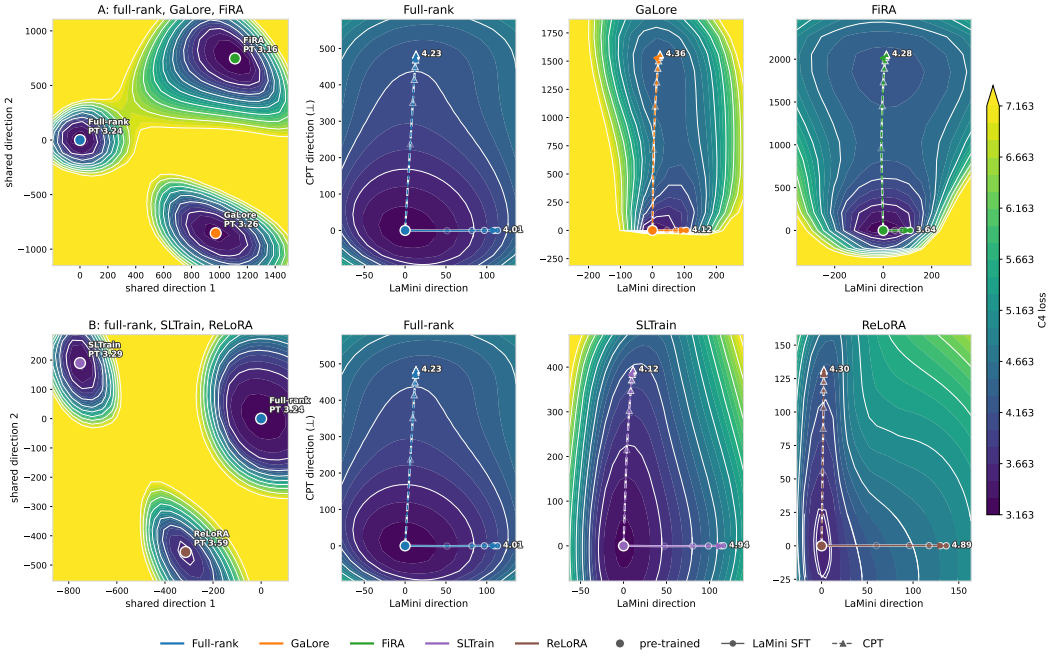


Figure 26: C4 loss on 2-D planes at 130M. Left column: planes through the pre-trained checkpoints of full-rank, GaLore, and Fira (A) and of full-rank, SLTrain, and ReLoRA (B). Other columns: for each method, the plane spanned by its own LaMini SFT direction (x) and the part of its CPT direction orthogonal to it (y). Circles mark the pre-trained checkpoint, solid lines the LaMini SFT path, and dashed lines the CPT path. Numbers give the C4 loss at the end of each path. Axes are not shared across panels.