# The Path Not Taken: RLVR Provably Learns Off the Principals

**Hanqing Zhu**[*,1,2] **, Zhenyu Zhang**[2] **, Hanxian Huang**[1] **, DiJia Su**[1] **, Zechun Liu**[1] **, Jiawei Zhao**[1]**,**
**Igor Fedorov**[1] **, Hamed Pirsiavash**[1] **, Zhizhou Sha**[1] **, Jinwon Lee**[1] **, David Z. Pan**[2]**,**
**Zhangyang (Atlas) Wang**[†,2] **, Yuandong Tian**[†,1] **, Kai Sheng Tai**[†,1]
[1]Meta AI [2]The University of Texas at Austin [†] Equal advisory contribution

## Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) reliably improves the reasoning performance of large language models, yet it appears to modify only a small fraction of parameters. We revisit this paradox and show that sparsity is a surface artifact of a **model-conditioned optimization bias**: for a fixed pretrained model, updates consistently localize to preferred parameter regions, highly consistent across runs and largely invariant to datasets and RL recipes. We mechanistically explain these dynamics with a **Three-Gate Theory**: Gate I (KL anchor) imposes a KL-constrained update; Gate II (model geometry) steers the step *off principal directions* into low-curvature, spectrum-preserving subspaces; and Gate III (precision) hides micro-updates in non-preferred regions, making the off-principal bias appear as sparsity. We then validate this theory and, for the first time, provide a parameter-level characterization of RLVR's learning dynamics: **RLVR learns off principal directions in weight space**, exhibiting minimal spectral drift, substantially smaller principal-subspace rotation, and off-principal update alignment, whereas SFT targets principal weights and distorts the spectrum.

Together, these results provide the first parameter-space account of RLVR's training dynamics, revealing clear regularities in how parameters evolve. **Crucially**, we show that RL operates in a distinct optimization regime from SFT, so directly adapting SFT-era parameter-efficient fine-tuning (PEFT) methods can be flawed, as evidenced by our case studies on advanced sparse fine-tuning and LoRA variants. We hope this work charts a path toward a white-box understanding of RLVR and the design of **geometry-aware, RLVR-native** learning algorithms, rather than repurposed SFT-era heuristics.

## 1 Introduction

Large Reasoning Models (LRMs), such as OpenAI-o3 (Jaech et al., 2024) and DeepSeek-R1 (Guo et al., 2025), have advanced the ability of large language models to solve complex mathematical and programming tasks. A key driver is large-scale Reinforcement Learning with Verifiable Rewards (RLVR), which uses simple, easy-to-verify rewards to incentivize complex, multi-step reasoning.

Yet, despite these advances, the mechanisms by which RL shapes model representations and behavior remain poorly understood. Given the substantial computational resources devoted to RL (relative to SFT) (xAI, 2025) and the emergence of striking new behaviors, one might naturally assume that such progress arises from significant parameter changes. However, recent evidence points in the opposite direction: RL induces *sparse* parameter updates, whereas SFT yields *dense* ones (Mukherjee et al., 2025). This counterintuitive finding reveals a paradox, *a high-cost, high-gain process that relies on surprisingly minimal weight modification.*

**Key observation.** We resolve this paradox by uncovering a deeper mechanism behind the apparent sparsity: a **persistent, model-conditioned optimization bias**. For a fixed pretrained model, this bias concentrates visible updates into a narrow, stable subset of parameters and remains strikingly

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: .

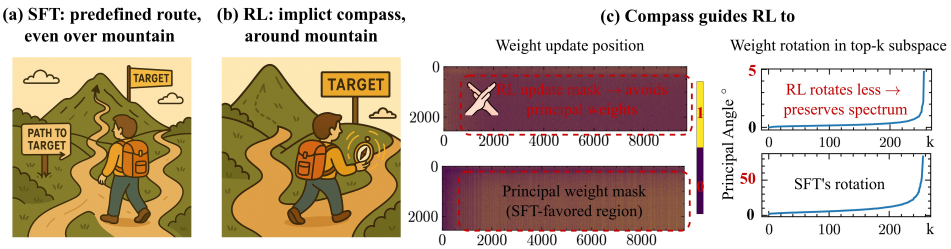


Figure 1: **SFT and RL follow distinct optimization paths.** (a) **SFT:** follows an externally guided route—even over the "mountain" (high curvature)—to reach the target. (b) **RLVR:** behaves as if steered by an *implicit compass*, detouring around the mountain (low curvature). (c) **Evidence.** *Left:* RL update positions versus principal-weight positions (largest-magnitude entries in rank-$k$ SVD of $W_0$ Liu et al. (2025c), details at Sec. 4.2); RL avoids principal regions while SFT targets them (Meng et al., 2024; Liu et al., 2025c). *Right:* principal-angle curves show RL rotates less (spectra preserved) and SFT rotates more.

invariant across diverse algorithms and datasets—*a model-conditioned feature*. `bfloat16` precision further accentuates the apparent sparsity by attenuating micro-updates in non-preferred regions. As illustrated in Fig. 1, we depict this bias as an *implicit compass*: unlike SFT with an explicit teacher, RLVR is subtly guided during optimization even without one.

**Research Question.** These phenomena raise a central question about RL's learning dynamics:

***Where does this optimization bias originate, and how does it shape parameter evolution?***

**Mechanistic explanation.** We formalize the mechanism behind RLVR's optimization dynamics through a **Three-Gate Theory**. Gate I (KL Anchor) enforces a KL-constrained update at each on-policy step. Gate II (Model Geometry) then steers this update *off the principal directions* toward low-curvature, spectrum-preserving subspaces embedded in the structured optimization landscape of a *pretrained* model, unlike training from a *randomly initialized* model. This geometry gate explains the *model-conditioned* nature of the bias: it arises from the pretrained landscape rather than particular datasets or RL recipes. Gate III (Precision) acts as a realization filter by hiding those micro-updates in non-preferred regions, making the off-principal bias *appear* sparse.

**Experimental validation.** We validate this theory with a comprehensive suite of experiments, uncovering **striking optimization dynamics**: *RLVR learns off the principal directions*, operating in a regime *disjoint* from SFT's. We show that (i) *RLVR preserves* the pretrained spectral structure with , whereas *SFT distorts* it; (ii) *RLVR avoids* principal weights—the high-energy directions indicated by rank-$k$ SVD reconstructions—whereas parameter-efficient *SFT targets* them (Liu et al., 2025c); and (iii) *RLVR depends on* the pretrained geometry: function-preserving orthogonal rotations *abolish* the effect of update locality overlap, consistent with a *model-conditioned* optimization bias.

**Rethinking Learning Algorithms for RLVR.** Beyond characterizing the optimization behavior, our findings show that RLVR operates in an optimization regime fundamentally distinct from SFT. Consequently, direct adaptations of SFT-era PEFT tricks may be flawed, especially those over-aligned with SFT's optimization dynamics. *(1) Sparse fine-tuning.* Restricting updates to *principal* weights, an SFT prior (Liu et al., 2025c), yields the worst optimization trend and severely degrades performance (per forward-KL drift (Shenfeld et al., 2025)). Conversely, updating non-principal, low-magnitude weights, as predicted by our theory, closely traces the dense RLVR trajectory. *(2) LoRA variants.* A concurrent report (Schulman & Lab, 2025) observes that low-rank LoRA (even rank-1) can match full-parameter performance in RL. However, our theory challenges their belief that advanced LoRA variants, **PiSSA** (Meng et al., 2024), may offer further gains. PiSSA targets principal weights, suited for SFT but fundamentally *misaligned with RLVR's off-principal dynamics*. We show PiSSA provides no additional gain over LoRA; worse, enforcing principal-direction updates *destabilizes training and leads to rapid collapse.*

**Contributions.** Our work makes the following key contributions:

- **Observation.** We identify a *persistent, data- and algorithm-invariant optimization bias* in RLVR fine-tuning, an *implicit optimization compass* that drives update behaviors.
- **Theory.** We propose the **Three-Gate Theory** (KL Anchor, Geometry, Precision) that explains how RL updates are constrained, steered, and filtered to produce the unique optimization pattern.

Table 1: **Update sparsity in SFT vs. RLVR.** *Higher* $\text{sparsity}_{\text{bf16}}$ indicates more weights unchanged. RLVR is consistently much sparser than SFT. † *Mixed* denotes a diverse data source combining math, coding, STEM, logic puzzles, and instruction-following Liu et al. (2025a).

| Base Model | FT Model | Algorithm | Data | $\text{sparsity}_{\text{bf16}}$ |
|---|---|---|---|---|
| Qwen-1.5B | DS-R1-Distill-Qwen-1.5B | SFT | Mixed | 2.8% |
| DS-R1-Distill-Qwen-1.5B | DeepScaleR-1.5B-Preview | GRPO | Math | 53.8% |
| DS-R1-Distill-Qwen-1.5B | DeepCoder-1.5B-Preview | GRPO | Code | 45.5% |
| DS-R1-Distill-Qwen-1.5B | Archer-Code-1.5B | GRPO | Code | 52.5% |
| DS-R1-Distill-Qwen-1.5B | NV-ProRL | GRPO | Mixed† | 38.4% |
| DS-R1-Distill-Qwen-1.5B | NV-ProRL-v2 | Reinforcement++ | Mixed† | 36.3% |
| Qwen3-8B-Base | Klear-Reasoner-8B-SFT | SFT | Math+Code | 0.6% |
| Klear-Reasoner-8B-SFT | Klear-Reasoner-8B | GRPO | Math+Code | 69.5% |
| Qwen3-8B-Base | GT-Qwen3-8B-Base | GRPO | Math | 79.9% |
| Qwen3-8B-Base | OURS | DAPO | Math | 79.7% |
| Qwen3-14B-Base | UniReason-Qwen3-14B-think-SFT | SFT | Math | 18.8% |
| Qwen3-14B-Base | UniReason-Qwen3-14B-RL | GRPO | Math | 68.3% |
| Qwen3-4B | Polaris-4B-Preview | DAPO | Math | 79.3% |
| DS-R1-Distill-Qwen-7B | Polaris-7B-Preview | DAPO | Math | 61.7% |
| Qwen3-30B-A3B | UloRL-A3B | GRPO | Math | 91.7% |

- **Evidence.** We present strong parameter-level validation consistently contrasting RL and SFT, including invariant spectra, low overlap with principal weights, and causal interventions that confirm geometry as the steering core of optimization.
- **Insight.** A principled basis for *rethinking* optimization strategies in RL-based post-training; we show that over-designed SFT-era sparse/low-rank priors (e.g., principal-targeted variants) are misaligned with RLVR's geometry-driven regime.

Our study provides the **first parameter-space account** linking RL optimization dynamics to weight evolution, complementing concurrent works that focus primarily on policy-level or distributional effects (Wu et al., 2025; Shenfeld et al., 2025). Crucially, our results reveal that RL operates in a distinct optimization regime from SFT, calling for rethinking RL-targeted PEFT recipes (see Sec. 5).

## 2 A Persistent, Model-Conditioned Optimization Bias in RLVR

While RL is known to induce sparse parameter updates, we ask *where* it localizes these changes. We identify a **model-conditioned optimization bias**: RL consistently routes updates to specific network regions. This bias is largely invariant to the dataset or RL algorithm, yet dependent on the base model. The observed sparsity is thus merely a *superficial readout* of this underlying mechanism.

**Model suite.** We analyze publicly released checkpoints, as shown in Tab. 1. The suite spans multiple RLVR variants (e.g., GRPO, DAPO, Reinforcement++), diverse data domains (math, coding, instruction), and several model families and types (dense and Mixture-of-Experts). We place particular emphasis on `DeepSeek-R1-Distill-Qwen-1.5B` (`DS-Qwen-1.5B`), for which a long-horizon RL checkpoint is available (Liu et al., 2025a). This model serves as a robust case study given its extensive training for over 3,000 steps on a diverse data mixture encompassing mathematics, coding, STEM, logic puzzles, and instruction-following tasks.

### 2.1 A Robust, `bfloat16`-aware Analysis of Update Sparsity

**A `bfloat16`-aware probe for unchanged weights.** `bfloat16` (bf16) is standard in modern RL frameworks like verl (Sheng et al., 2024), to improve throughput without compromising performance. However, analyzing parameter changes under bf16 requires a careful probe. Its unique numerical format, with only 7 mantissa bits for precision, means that the smallest representable difference between two numbers scales with their magnitude. Consequently, a fixed absolute-tolerance check as used in (Mukherjee et al., 2025), is *unreliable*, which can over- or under-report the fraction of unchanged weights (see Appendix F.1).

To ensure a rigorous report, we adopt a numerically robust, `bfloat16`-aware probe to define the update sparsity $\text{sparsity}_{\text{bf16}}$ as the fraction of parameters that remain unchanged.

**Definition 2.1** (Unchanged Weight in bf16)**.** *Let $w_i, \widehat{w}_i \in \mathbb{R}$ be scalars stored in bf16 (finite, nonzero). We say $w_i$ is* unchanged *with respect to $\widehat{w}_i$ iff*

$$\left|\widehat{w}_i - w_i\right| \le \eta \max\left(|w_i|, |\widehat{w}_i|\right), \qquad \eta = 10^{-3}. \tag{1}$$

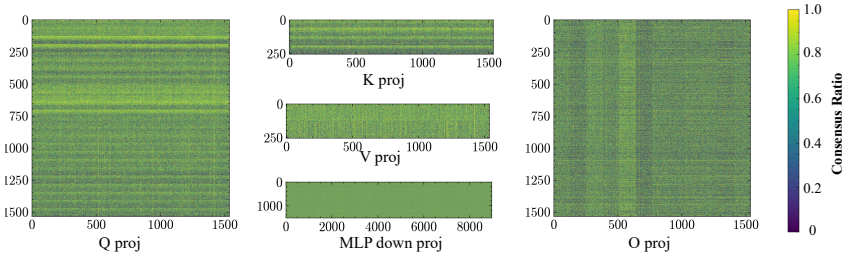


Figure 2: **Consensus ratio of weight updates** across five RLVR runs on the 13th layer's projection (Q/K/V/O) and the MLP down projection (zoom in for structures). Lighter bands indicate coordinates updated in most runs, revealing a stable, stripe-like routing pattern rather than random scatter.

*Choosing* $\eta$=$10^{-3} < 2^{-9}$ *makes equation 1 equivalent to bitwise equality (See Appendix F.2,).*

**Definition 2.2** (bf16-aware Update Sparsity)**.** *Write* $x \approx_\eta^{\mathrm{bf16}} y$ *for Def. 2.1. Define the bf16 change count* $\|\theta^1 - \theta^0\|_{0,\eta}^{\mathrm{bf16}} := \big|\{\, i : \theta_i^1 \not\approx_\eta^{\mathrm{bf16}} \theta_i^0 \,\}\big|$ *and the corresponding sparsity*

$$\mathrm{sparsity}_{\mathrm{bf16}}(\theta^0, \theta^1; \eta) := 1 - \|\theta^1 - \theta^0\|_{0,\eta}^{\mathrm{bf16}} / n. \tag{2}$$

*where* $n$ *is the total number of parameters. Values near* 1 *indicate few stored changes, while values near* 0 *indicate dense apparent change.*

**RLVR update sparsity results.** As shown in Tab. 1, our analysis confirms that RL yields substantially higher update sparsity than SFT. Across models, SFT sparsity is consistently low (typically 0.6%–18.8%), whereas RL sparsity is an order of magnitude higher, ranging from 36% to 92%. However, absolute levels on recent checkpoints are lower than earlier reports (Mukherjee et al., 2025), underscoring the need for bf16-aware probes and re-evaluation on current models.

### 2.2 RLVR Exhibits Model-Conditioned Update Locality

Magnitude alone does not reveal *where* changes occur, impeding the deep analysis on *how* sparse changes arise. We therefore examine the *updated subnetwork*. We use 5 independent RLVR checkpoints from the same `DS-Qwen-1.5B` in Tab. 1, trained on diverse data and different RLVR algorithms. For each layer $\ell$ and run $r$, we first form the bf16-aware *changed* mask $M_\ell^{(r)} := \mathbf{1}\big[\, W_\ell^{(r)} \not\approx_\eta^{\mathrm{bf16}} W_\ell^0 \,\big]$ (Def.2.2) against the base weights $W_\ell^0$.

Table 2: Cross-run stability for 13th block.

| Layer | Jaccard Overlap | Random Baseline |
|---|---|---|
| Q | 0.580 | 0.430 |
| K | 0.580 | 0.413 |
| V | 0.597 | 0.467 |
| O | 0.552 | 0.373 |
| MLP-down | 0.585 | 0.453 |
| MLP-up | 0.578 | 0.443 |
| MLP-gate | 0.575 | 0.437 |

**Stability across runs.** We first analyze their spatial agreement using *Jaccard Overlap*. For runs $r, s$, let $A = \{(i,j) : M_{\ell,ij}^{(r)} = 1\}$ and $B = \{(i,j) : M_{\ell,ij}^{(s)} = 1\}$. We report the mean off-diagonal of the pairwise Jaccard matrix $J(A,B) = \frac{|A \cap B|}{|A \cup B|}$ and compare it to the independent Bernoulli baseline $\mathbb{E}[J] = \frac{pq}{p+q-pq}$. As summarized in Tab. 2, Jaccard is consistently high across runs, confirming a shared footprint when trained from the same base model, with Jaccard matrix shown in Fig. 10.

**Consensus ratio (*where updates land*).** Stability alone does not indicate *where* updates land. We therefore visualize and analyze the consensus ratio $C_{\ell,ij} = \frac{1}{R}\sum_{r=1}^{R} M_{\ell,ij}^{(r)}$, the fraction of runs realizing a *weight update* at coordinate $(i,j)$. Values near 1 indicate that *all* runs consistently change that weight; values near 0 indicate that none do. As shown in Fig. 2, consensus maps reveal contiguous row/column bands, stripe-like, localized routing rather than scattered noise. Especially, there are obvious *row-wise stripes in* Q/K/V *projections and column-wise stripes in* O *projections. This exposes a clear* ***optimization bias****: RLVR consistently concentrates updates in specific regions of the parameter matrices, even though the five runs use disjoint data and RL variants.*

**Temporal stability (*how the bias emerges*).** To examine *within-run* dynamics, we track the row-wise ratio $\rho_{\ell,i}(t) = \frac{1}{n_\ell}\sum_j M_{\ell,ij}(t)$ and column-wise ratio $\kappa_{\ell,j}(t) = \frac{1}{m_\ell}\sum_i M_{\ell,ij}(t)$ across checkpoints at $t$ steps. On `DS-Qwen-1.5B` (training setting in Appendix D.1), the *relative* profiles $\rho_{\ell,\cdot}(t)$ and $\kappa_{\ell,\cdot}(t)$ remain aligned while overall density grows as shown in Fig. 3: peaks and troughs

persist. The routing bias *emerges early* and is *reinforced over training*, indicating a temporally stable phenomenon rather than a transient artifact. Moreover, the peak is consistent with the bias structure shown in Fig. 2. We also show their remaining column-wise (Q) and row-wise (O) update ratio dynamics in Fig. 12, without a clear trend, *indicating the bias is indeed structured, not random.*

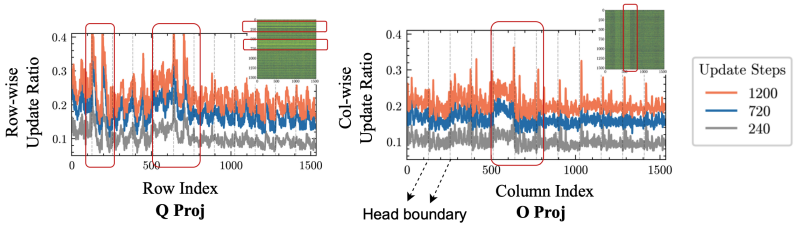


Figure 3: **Temporal emergence of the optimization bias** with row and column-wise update ratios for the 13th attention block across gradient update steps ($t \in \{240, 720, 1200\}$), smoothed with a 3-step window. The row-dominant (Q) and column-dominant (O) patterns are consistent with the bias structures in Fig. 2. We visualize the head boundaries with grey dashed lines. The bias appears not only across heads but also within heads.

**Other model families (*whether only on Qwen*).** We observe similar stripe-structured footprints on `Llama` and `Mistral` (Fig. 11 in Appendix), suggesting the routing bias is generic to RLVR.

### 2.3 Sparsity Is a Superficial Artifact of the Optimization Bias

The stable footprint of where updates land, persisting both throughout training and in the final model, suggests the focus should move from sparsity itself to the underlying optimization bias.

We find that sparsity is actually the *readout* of this optimization bias, whose visibility is amplified by the precision limits of `bf16` storage. Because `bf16` has a limited mantissa, changes smaller than the unit-in-the-last-place (ULP) threshold (Lemma F.2) are not representable. *Therefore, if RLVR consistently routes sub-ULP updates toward a particular subset of parameters, the stored values will not change, and the result appears as sparsity.*

We test this hypothesis by increasing the learning rate to scale otherwise sub-ULP updates above the representable threshold. As predicted, the apparent update sparsity largely disappears. This directly challenges the interpretation of (Mukherjee et al., 2025) that sparsity stems from zero gradients. Instead, our results point to sparsity as a byproduct of an optimization bias interacting with finite precision. Consistent with this view, concurrent work observes that sparsity mostly vanishes under `fp32` storage (Shenfeld et al., 2025), even though task performance does not improve.

**Remark on precision.** One natural confusion is treated the bf16 as the final cause, while it is important to note that in verl, optimizer states and gradient reductions/accumulation are maintained in `float32`[1]. So the sparsity cannot show up unless the RL process is consistently biased toward where to assign visible changes throughout the training.

***Aha Finding! — RLVR exhibits a patterned, rather than random, optimization bias toward where the visible changes land. The sparsity is a direct readout of this underlying bias.***

## 3 A Mechanistic Theory of RL's Unique Optimization Dynamics

In the post-training era, RL has become a key stage, albeit with intensive compute (xAI, 2025). Paradoxically (Sec. 2), these gains arise not from broad parameter changes but from selective, patterned edits that reveal a persistent optimization bias. Understanding this distinctive training behavior raises the central question:

***Where does this optimization bias originate, and how does it shape parameter evolution?***

We characterize these optimization dynamics with the *Three-Gate Theory*, KL Anchor, Model Geometry, and Precision, which mechanistically explains how on-policy RL updates are *constrained* via Gate I (KL Anchor; Sec. 3.1), *steered* via Gate II (Model Geometry; Sec. 3.2), and *filtered* via Gate III (Precision; Sec. 3.3) into the observed update pattern.

[1] verl mixed-precision settings with {`reduce_type, buffer_dtype`}=`float32`.

**Notations.** We consider a large language model with parameters $\theta$, defining a conditional distribution $\pi_\theta(y \mid x)$ over possible output token sequences $y = (y_1, \ldots, y_T) \in \mathcal{Y}$ given a prompt $x \in \mathcal{X}$ from the space $\mathcal{X}$. Each sequence $y$ is composed of tokens from a vocabulary $\mathcal{V}$ of size $N$.

### 3.1 Gate I: On-Policy RL Imposes a One-Step KL Leash

We first show that online policy gradient updates yield a per-step *policy* KL bound (an ***anchoring*** effect), which in turn limits parameter movement during the RLVR update.

**RLVR objective.** Various RLVR algorithms including PPO, GRPO, DAPO, and REINFORCE++, learn a policy $\pi_\theta$ by optimizing variants of a KL-regularized objective:

$$\max_\theta \mathbb{E}_{y\sim\pi_\theta(\cdot|x),x\sim\mathcal{X}}[R(x,y) - \beta \mathrm{KL}(\pi_\theta(\cdot \mid x) \,\|\, \pi_{ref}(\cdot \mid x))]. \tag{3}$$

where $\pi_{\text{ref}}$ is a fixed reference policy and $\beta \geq 0$ controls the KL regularization ($\beta = 0$ recovers the clip-only variants such as DAPO). Rewards $R(x, y)$ are *verifiable* and (after normalization) *bounded* (e.g., pass/fail or execution scores). Moreover, the surrogate typically uses the token-wise importance ratio $w_t = \frac{\pi_\theta(y_t|x,y_{<t})}{\pi_{\text{old}}(y_t|x,y_{<t})}$ with clipping relative to $\pi_{\text{old}}$.

**One-step surrogate.** With equation 3, a standard sequence-level online policy-gradient surrogate is

$$\mathcal{L}_{\text{PG}}(\theta) = -\mathbb{E}_{x\sim\mathcal{X},\, y\sim\pi_\theta(\cdot|x)}\big[A^\perp(x,y)\, \log \pi_\theta(y \mid x)\big], \tag{4}$$

where $A^\perp$ is a (normalized) advantage estimate, optionally *shaped* by a reference-KL log-ratio term. In practice, updates are performed over mini-batches, with a collected batch of data, not in a fully on-policy manner. But the resulting error after a small step size $\Delta\theta$ is $O(\|\Delta\theta\|^2)$ (Lemma G.1).

**Implicit KL leash.** The KL leash emerges as policy gradient methods can be understood as a conservative projection, keeping new policy close to its starting point while reweighting it toward higher-reward outcomes, not pulling it toward a potentially distant external distribution like SFT:

**Proposition 3.1** (One-step policy-KL leash)**.** *Let $q(\cdot \mid x)$ be a full-support reference and let $\tilde{q}_\beta(\cdot \mid x) \propto q(\cdot \mid x) \exp(R/\beta)$ denote the soft-regularized improvement oracle. Let $\theta^+$ be the parametric fit obtained by the $M$-projection of $\tilde{q}_\beta$ onto the policy class, $\theta^+ \in \arg\min_\theta D_{\mathrm{KL}}(\tilde{q}_\beta \| \pi_\theta)$. Then, for a sufficiently small one-step update,*

$$D_{\mathrm{KL}}\big(\pi_{\theta^+} \,\|\, \pi_\theta\big) \;\leq\; (1 + o(1))\, D_{\mathrm{KL}}\big(\tilde{q}_\beta \,\|\, \pi_\theta\big), \tag{5}$$

*where the $o(1)$ term vanishes as $D_{\mathrm{KL}}(\tilde{q}_\beta \| \pi_\theta) \to 0$.*

Notably, even when the explicit KL term is removed (e.g., in DAPO with $\beta = 0$), the ratio clipping trick still imposes a KL bound $O(\varepsilon^2)$ in the small-step regime (Appendix. G.2.4), confirmed empirically with a bounded KL divergence change during a DAPO run (Fig. 13).

**Weight update constraint.** Now we show the KL leash puts a constraint on weight update $\Delta W$

**Proposition 3.2** (Policy-KL leash $\Rightarrow$ weight bound)**.** *Assume $\log \pi_\theta$ is $C^3$ and let $F(\theta)$ denote the Fisher information. If a one-step update $\theta^+ = \theta + \Delta$ satisfies $D_{\mathrm{KL}}(\pi_{\theta^+} \| \pi_\theta) \leq K$ and, on the update subspace, $F(\theta) \succeq \mu I$ for some $\mu > 0$, then for $K$ sufficiently small*

$$\|\Delta\|_{F(\theta)} \triangleq \sqrt{\Delta^\top F(\theta) \Delta} \;\leq\; \sqrt{2K}\,(1 + o(1)), \qquad \|\Delta\|_2 \;\leq\; \sqrt{\tfrac{2K}{\mu}}\,(1 + o(1)). \tag{6}$$

*Consequently, for any weight matrix block $W \subset \theta$, $\|\Delta W\|_F \leq \sqrt{2K/\mu}\,(1 + o(1))$.*

See a detailed proof for Proposition 3.1 in Appendix G.2.1 and Proposition 3.2 in Appendix G.2.2.

> ***Take-away 1: RL update imposes an implicit KL leash (anchor effect), ensuring that the per-step drift from the current policy is small.*** This aligns with recent work arguing that even the final policy is KL-proximal (Wu et al., 2025; Shenfeld et al., 2025). Our focus, however, is to understand how this leash affects the weight change dynamics.

### 3.2 Gate II: Model Geometry Determines *Where* a KL-Bounded Step Goes

**From Gate I to *location*.** Gate I supplies a one-step KL leash that bounds the move, but it does not specify *where* the update lands. We propose Gate II(Model Geometry), where we argue that, unlike

a randomly initialized network, a well-pretrained model possesses a highly structured geometry, e.g., spectral statistics and high-curvature directions during optimization, that determines where a KL-constrained update goes.

**Layerwise norm bound from the KL leash.** Let $W_0$ be a pretrained linear block, $W_+ = W_0 + \Delta W$ the post-step block, and let $S_W \succeq \mu_W I$ be a per-layer curvature proxy . If the per-layer KL budget satisfies $\frac{1}{2}\langle \mathrm{vec}\,\Delta W, S_W\ \mathrm{vec}\,\Delta W\rangle \le \delta_W$, then (Appendix G.10)

$$\|\Delta W\|_F \le \sqrt{\tfrac{2\delta_W}{\mu_W}}, \qquad \|\Delta W\|_2 \le \sqrt{\tfrac{2\delta_W}{\mu_W}}. \tag{7}$$

We then show this conservative update yields three consequences making them preserve pretrained weight spectrum instead of destroying them based on weight perturbation theory (Stewart, 1998).

**Limited subspace rotation.** First, as shown in Theorem 3.3, the angle between the original and updated subspaces is quadratically bounded, meaning the fundamental directions are preserved.

**Theorem 3.3** (Constrained subspace rotation with Wedin's sin–$\Theta$ theorem (Wedin, 1972))**.** *For any $k$ with $\gamma_k > 0$,*

$$\max\Big(\big\|\sin\Theta(U_k(W_0), U_k(W_+))\big\|_2,\ \big\|\sin\Theta(V_k(W_0), V_k(W_+))\big\|_2\Big) \le \frac{\|\Delta W\|_2}{\gamma_k} \le \frac{\sqrt{2\delta_W/\mu_W}}{\gamma_k}. \tag{8}$$

**Singular value stability.** Second, the magnitudes of the principal components themselves are preserved. The change in each singular value is bounded by the norm of the update.

**Corollary 3.4** (Singular-value stability)**.** *For each $k$,*

$$|\sigma_k(W_+) - \sigma_k(W_0)| \le \|\Delta W\|_2 \le \sqrt{\frac{2\delta_W}{\mu_W}}, \qquad \sum_i \big(\sigma_i(W_+) - \sigma_i(W_0)\big)^2 \le \|\Delta W\|_F^2 \le \frac{2\delta_W}{\mu_W}. \tag{9}$$

**Top-k energy preservation.** Finally, these effects combine to ensure the cumulative energy of the top-k components of the weights remains stable.

**Corollary 3.5** (Top-$k$ energy and Ky Fan norms)**.** *Let $\|\cdot\|_{(k)} := \sum_{i=1}^k \sigma_i(\cdot)$ be the Ky Fan $k$-norm. Then*

$$\big|\,\|W_+\|_{(k)} - \|W_0\|_{(k)}\,\big| \le \sum_{i=1}^k \big|\sigma_i(W_+) - \sigma_i(W_0)\big| \le k\,\|\Delta W\|_2 \le k\sqrt{\tfrac{2\delta_W}{\mu_W}}. \tag{10}$$

See a detailed proof in Appendix G.3.

***Take-away 2: Under the KL leash, RL updates tend to preserve the model's original weight structure rather than destroy it. This naturally favors updates in low-curvature directions of the optimization landscape, which avoids dramatic changes in model behavior.*** Since directly quantifying curvature in LRM with long CoTs is computationally prohibitive, we instead adopt a powerful and efficient proxy, principal weights (Liu et al., 2025c), as detailed in Sec. 4.2.

### 3.3 Gate III: Precision Acts as a Lens Revealing the Compass

Building on the optimization bias, the bfloat16 with limited precision acts as a *lens*: it hides those micro-updates that occur where the RL consistently holds a weak willingness to apply large changes.

**Corollary 3.6** (Magnitude-dependent realization threshold)**.** *A stored weight $W_{ij}$ changes at a step iff $|\Delta W_{ij}| \gtrsim \frac{1}{2}\,\mathrm{ULP}_{\mathrm{bf16}}(W_{ij})$.*

The effect of this gate has been discussed aforementioned. We would emphasize again that precision is more an *amplifier* for visible sparsity, not the *cause* of optimization bias, as optimizer states, etc., are still in float32 (See Sec. 2.3).

## 4 Theory-Guided Validation of RLVR's Optimization Dynamics

We conduct theory-guided experiments analyzing how RLVR modifies parameters and interacts with pretrained geometry. These results validate our central prediction: the pretrained model geometry steers *KL-constrained* updates, yielding *distinct, off-principal optimization dynamics* that set RLVR apart from SFT.

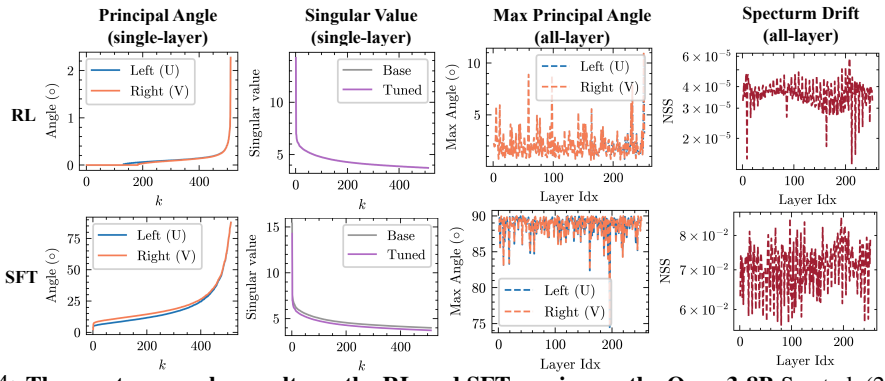


Figure 4: **The spectrum probe results on the RL and SFT version on the Qwen3-8B** Su et al. (2025). For the same exemplar layer, we display top-$k$ principal angles and singular value curves; the right panels report the maximum principal angles and spectrum drift across all layers. RLVR maintains a stable top-$k$ spectrum with minimal subspace rotation, unlike SFT.

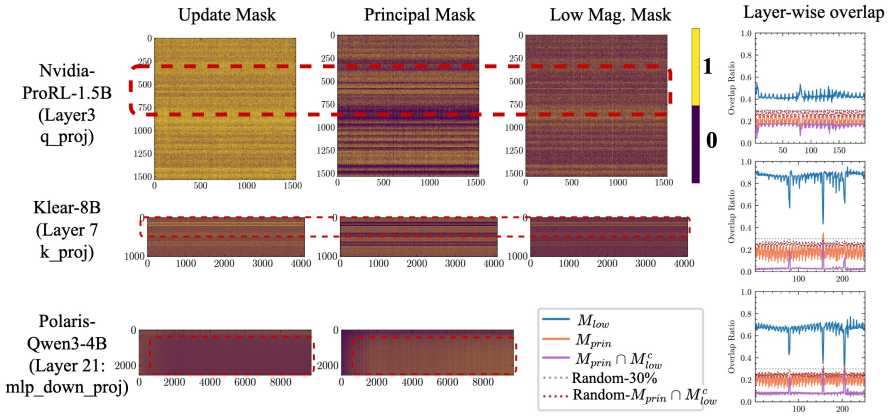


Figure 5: **RL avoids updating principal weights.** We compare the RL update mask with principal weight mask $M_{princ}$, low magnitude mask $M_{low}$, and the one $M_{princ} \cap M_{low}^c$. The layer-wise overlap between RL updates and principal weights is consistently *sub-random*, an effect more pronounced when removing its overlapped weights with $M_{low}$, i.e., $M_{princ} \cap M_{low}^c$.

### 4.1 RLVR Preserves Spectral Geometry, While SFT Distorts It

We begin by probing spectral changes to test whether RL updates are steered toward low-curvature, spectrum-preserving directions. If so, RLVR should largely preserve the pretrained spectral structure, whereas SFT, lacking this steering, should significantly distort it.

**Setups.** We analyze checkpoints from a standard SFT→RLVR pipeline on Qwen3-8B-Base (Su et al., 2025) and a long-horizon RL run on `DS-Qwen-1.5B` (Liu et al., 2025a). We also consider a setting where SFT and RL are applied separately to Qwen3-14B-Base, matched on in-domain math performance (Huan et al., 2025). In all cases, we compare base weights $W_0$ and fine-tuned weights $W_+$.

**Metrics.** We compare the base weights $W_0$ with the finetuned weights $W_+$:

- **Subspace rotation.** For the top-$k$ left ($U$)/right($V$) singular subspaces, we check the rotation using **principal angles** via $\cos\theta_i(U) := \sigma_i\left(U_{0,k}^\top U_{+,k}\right)$ and $\cos\theta_i(V) := \sigma_i\left(V_{0,k}^\top V_{+,k}\right)$.
- **Spectrum drift.** Beyond showing the singular value curve, we quantify singular-value change with a normalized $\ell_2$ shift: $\mathrm{NSS}(W) = \|\sigma(W_+) - \sigma(W_0)\|_2 / \|\sigma(W_0)\|_2$

**Our findings.** RLVR checkpoints exhibit a *Insightably stable* spectrum within the top principal components: across layers, RLVR shows *consistently small* principal-subspace rotation and *minimal* spectral drift. The singular-value profiles are *even nearly identical* to the base model. By contrast, SFT induces *substantially larger* rotations and *pronounced* drifts on the same metrics (Fig. 4).

### 4.2 RLVR Avoids Principal Weights, While SFT Targets Them

We now move from macro-level spectral analysis to a micro-level examination of individual weights, probing *which parameters RLVR favors or avoids to update*, a deeper investigation into the parameter-space dynamics.

**Principal weights as a proxy for high-curvature directions.** Directly identifying high-curvature directions is computationally prohibitive, especially given LRM with long CoTs. Instead, we adopt a powerful proxy from recent work Liu et al. (2025c), ***principal weights*****,** which is defined as *the weights with the largest magnitude after low-rank approximation*, representing its most influential computational pathways. The validity of this proxy is confirmed by their perturbation studies, which show that modifying these specific weights causes sharp *reasoning performance degradation*. This degradation is directly linked to high-curvature regions via a Taylor expansion of the loss. The *principal mask*, $M_{\text{princ}}^{(k)} = \text{Top}_\alpha\big(s_{ij}^{(k)}\big)$, is defined as the top-$\alpha$ fraction of weights with the highest score, $s_{ij}^{(k)} = |W_0^{(k)}(i,j)|$, where $W_0^k$ is the rank-$k$ SVD reconstruction of $W_0$.

**Low-magnitude weights as low-resistance pathway.** We further include the top-$\alpha$ lowest magnitude weights, as $M_{\text{low}} = \text{Bottom}_\alpha\big(|W_0|\big)$. The magnitude is also a bias from the model geometry (distribution prior), impacting how easily the weights can be updated based on our precision gate.

**Metrics.** Let $M$ be the weight update *update mask* from an RLVR run. We report the overlap ratio between our identified mask $M_\bullet$ with it, defined as $\text{Overlap}(M_\bullet, M) = \frac{|M_\bullet \cap M|}{|M|}$., with a random guess baseline overlap ratio as the density of $M_\bullet$ itself., i.e., $\alpha$.

**Our findings.** Fig. 5 visualizes the RL update mask $M$ in relation to the principal mask $M_{princ}$ and the low-magnitude mask $M_{low}$, reporting their layer-wise overlap against a random baseline as well. The results show a clear dichotomy. RL updates exhibit a sub-random overlap with principal weights, indicating a strong tendency to avoid them. Conversely, the updates show a super-random overlap with low-magnitude weights due to their low-resistance to micro-updates. Besides, we found that the residual overlap between updates and principal weights is highly accounted for by weights that are both principal (defined by the rank-k approximation of $W_0$) and low-magnitude (original $W_0$). After excluding this intersection, i.e., $M_{princ} \cap M_{low}^c$, the overlap drops significantly.

**Insight.** This points to a central implication: *RLVR and SFT operate in distinct optimization regions of parameter space*, even at comparable task performance. *RLVR avoids high-curvature, principal regions, whereas SFT targets them.* This regional mismatch helps explain the limited transferability of SFT-oriented PEFT under RL (Sec. 5).

### 4.3 RLVR Relies on Model Geometry, Disrupting Geometry Destroys the Bias

Gate II posits that the pretrained model's geometry steers RL updates. To test this causal link, we deliberately "scramble" the geometry of specific layers in a Qwen3-4B-Base model using orthogonal rotations for O/V layers (ROTATE) and head permutations for all Q/K/V/O layers (PERMUTE) (details in Appendix E) and compare the update overlap ratio $\text{Overlap}(M_\bullet, M) = \frac{|M_\bullet \cap M|}{|M|}$. between the base run with another independent run without intervention and one run with intervention.

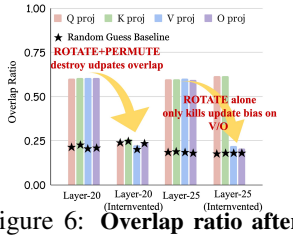


Figure 6: **Overlap ratio after intervention.**

**Our Findings.** We modify (i) layer 20 with ROTATE+PERMUTE, and (ii) layer 25 with ROTATE. As shown in Fig. 6, the update overlap collapsed to a random level in the intervened layers, while remaining high in all untouched layers. This provides strong causal evidence that the pretrained model's geometry is the source of the optimization bias.

## 5 Theory-Guided Rethinking of RL Learning Algorithms

A good theory should not only *explain* observations but also *inform* design. Our account shows that RLVR and SFT follow disjoint parameter-space dynamics, implying that successful SFT-era efficient PEFT variants, especially those over-aligned with principal directions via sparse or low-rank priors,

may fail to transfer to RLVR. This section both *validates* our predictions and *demonstrates* how they inform the redesign of learning algorithms for RL. We put the LoRA study at Appendix A.

### 5.1 Probing Sparse Fine-Tuning in RL

We construct a **parameter mask** identified *without any additional training* and apply it to perform sparse RL fine-tuning. Following (Shenfeld et al., 2025), we track the token-wise forward KL divergence $\mathrm{KL}(\pi \,\|\, \pi_{\text{ref}})$ between the fine-tuned policy and the base model throughout training. This metric quantifies how closely a sparse run follows the dense baseline trajectory, if pruning certain weights impedes learning, the KL drift will slow, indicating blocked optimization progress.

**Mask design.** We evaluate several masks constructed directly from the pretrained model: (i) $M_{\text{princ}}$ (*principal-only*, top-50% principal weights), (ii) $M^c_{\text{princ}}$ (*non-principal-only*, the complementary subspace), (iii) $M_{\text{low}}$ (*low-magnitude-only*), (iv) $M_{\text{low}} \cup M^c_{\text{princ}}$ (*safe mask*, favoring non-principal and low-magnitude weights), , and (v) a random mask with the same layer-wise sparsity as (iv). We choose 50% for (i) as we want to isolate the effect of the number of parameters for a fair comparison to see the difference between (i) and (ii).

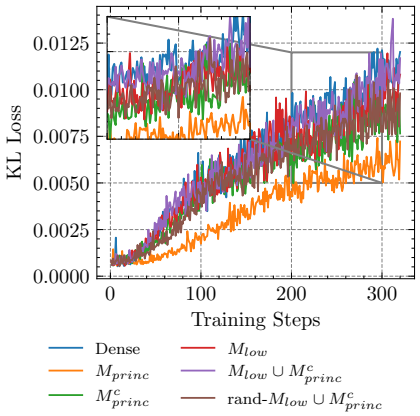


Figure 7: **KL loss curves on** `DS-Qwen-1.5B` under different masks.

**Our findings.** (KL in Fig. 7 and accuracy in Tab. 3) As shown in Fig. 7, the **safe mask** $M_{\text{low}} \cup M^c_{\text{princ}}$ most closely tracks the dense RLVR run's KL curve and achieves comparable final accuracy, indicating that our theory correctly identifies *highly touchable weights*. In contrast, the **principal-only mask** yields the **worst optimization trend**-its KL curve rises slowly, showing excessive intervention and degraded training dynamics. This directly confirms that the directions favored by SFT (principal weights) are ineffective for RL.

**Insight.** Our results suggest a simple yet effective alternative: *Freezing principal and large-magnitude weights while updating non-principal, low-magnitude ones* closely reproduces dense RLVR behavior (KL trajectory and final accuracy) using roughly 70% the parameters This shows that our theory provides **practical guidance** for identifying the effective subspace of RL updates, *entirely without additional training*. While the masks used here are *one-shot* and fixed, combining this framework with dynamic mask refresh or adaptive scheduling (Zhao et al., 2024; Zhu et al., 2024; Liu et al., 2025c) is a promising next step.

**Takeaway. RLVR operates in a distinct, geometry-driven optimization regime, so *your old PEFT tricks may not work.***

## 6 Conclusion

We revisited the paradox of *visible update sparsity* in RLVR and showed that it is a superficial readout of a deeper, *model-conditioned, geometry-aligned optimization bias* that determines where updates land. We formalized this mechanism with the **Three-Gate Theory**: a KL anchor constrains each on-policy step; pretrained geometry steers updates *off principal directions* into low-curvature, spectrum-preserving subspaces; and finite precision renders the bias visible as sparsity by masking micro-updates. Empirically, RLVR preserves spectral structure and avoids principal weights, whereas SFT targets principal directions and distorts the spectrum; when the pretrained geometry is disrupted, these signatures vanish, establishing geometry as the steering core. Beyond explanation, our case studies bridge mechanism and practice: SFT-era principal-aligned PEFT (e.g., sparse/low-rank variants) often misaligns with RLVR's off-principal regime. Taken together, these results provide the *first parameter-level account* of RLVR's training dynamics, replacing a black-box view with a *white-box* understanding of how parameters evolve under RLVR, and laying the foundation for **geometry-aware, RLVR-native** parameter-efficient learning algorithms.

## Acknowledgment

We thank Zhengqi Gao (Massachusetts Institute of Technology) for insightful discussion on the framework and idea discussion.

## References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL `https://arxiv.org/abs/2501.12948`.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.

Daya Guo, Dejian Yang, Haowei Zhang, and Junxiao Song. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.07570*, 2025. URL `https://arxiv.org/abs/2501.07570`.

Seungwook Han, Jyothish Pari, Samuel J Gershman, and Pulkit Agrawal. General reasoning requires learning to reason from the get-go. *arXiv preprint arXiv:2502.19402*, 2025.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

Yafei Hu, Quanting Xie, Vidhi Jain, Jonathan Francis, Jay Patrikar, Nikhil Keetha, Seungchan Kim, Yaqi Xie, Tianyi Zhang, Hao-Shu Fang, et al. Toward general-purpose robots via foundation models: A survey and meta-analysis. *arXiv preprint arXiv:2312.08782*, 2023.

Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. *arXiv preprint arXiv:2507.00432*, 2025.

Aaron Jaech et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, Jianfeng Gao, et al. Multimodal foundation models: From specialists to general-purpose assistants. *Foundations and Trends® in Computer Graphics and Vision*, 16(1-2):1–214, 2024.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025a.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025b.

Zihang Liu, Tianyu Pang, Oleg Balabanov, Chaoqun Yang, Tianjin Huang, Lu Yin, Yaoqing Yang, and Shiwei Liu. Lift the veil for the truth: Principal weights emerge after rank reduction for reasoning-focused supervised fine-tuning. *arXiv preprint arXiv:2506.00772*, 2025c.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Michael Luo, Sijun Tan, Roy Huang, Xiaoxiang Shi, Rachel Xin, Colin Cai, Ameen Patel, Alpay Ariyak, Qingyang Wu, Ce Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepcoder: A fully open-source 14b coder at o3-mini level. https://pretty-radio-b75.notion.site/DeepCoder-A-Fully-Open-Source-14B-Coder-at-O3-mini-Level-1cf81902c14680b3bee5eb349a512a51, 2025a. Notion Blog.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025b. Notion Blog.

MAA. American mathematics contest 12 (amc 12), November 2023. URL `https://artofproblemsolving.com/wiki/index.php/AMC_12_Problems_and_Solutions`.

MAA. American invitational mathematics examination (aime), February 2024. URL `https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions`.

MAA. American invitational mathematics examination (aime), February 2025. URL `https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions`.

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

Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. *Advances in Neural Information Processing Systems*, 2025.

Long Ouyang et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *arXiv preprint arXiv:2303.08774*, 2018.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Negin Raoof, Etash Kumar Guha, Ryan Marten, Jean Mercat, Eric Frankel, Sedrick Keh, Hritik Bansal, Georgios Smyrnis, Marianna Nezhurina, Trung Vu, Zayne Rea Sprague, Mike A Merrill, Liangyu Chen, Caroline Choi, Zaid Khan, Sachin Grover, Benjamin Feuer, Ashima Suvarna, Shiye Su, Wanjia Zhao, Kartik Sharma, Charlie Cheng-Jie Ji, Kushal Arora, Jeffrey Li, Aaron Gokaslan, Sarah M Pratt, Niklas Muennighoff, Jon Saad-Falcon, John Yang, Asad Aali, Shreyas Pimpalgaonkar, Alon Albalak, Achal Dave, Hadi Pouransari, Greg Durrett, Sewoong Oh, Tatsunori Hashimoto, Vaishaal Shankar, Yejin Choi, Mohit Bansal, Chinmay Hegde, Reinhard Heckel, Jenia Jitsev, Maheswaran Sathiamoorthy, Alex Dimakis, and Ludwig Schmidt. Automatic evals for llms, 2025. URL `https://github.com/mlfoundations/evalchemy`.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

John Schulman and Thinking Machines Lab. Lora without regret. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250929. https://thinkingmachines.ai/blog/lora/.

Zhihong Shao et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*, 2025.

Guangming Sheng et al. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*, 2024.

Gilbert W Stewart. Perturbation theory for the singular value decomposition. 1998.

Zhenpeng Su, Leiyu Pan, Xue Bai, Dening Liu, Guanting Dong, Jiaming Huang, Wenping Hu, and Guorui Zhou. Klear-reasoner: Advancing reasoning capability via gradient-preserving clipping policy optimization. *arXiv preprint arXiv:2508.07629*, 2025.

Qwen Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Per-Åke Wedin. Perturbation bounds in connection with singular value decomposition. *BIT Numerical Mathematics*, 12(1):99–111, 1972.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

Fang Wu, Weihao Xuan, Ximing Lu, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why rlvr may not escape its origin. *arXiv preprint arXiv:2507.14843*, 2025.

xAI. Grok: Ai assistant, 2025. URL `https://x.ai/grok`. Accessed: 2025-09-24, continuously updated.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025a.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025b. URL `https://arxiv.org/abs/2503.18892`.

Simon Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Peter Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, et al. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. *Advances in neural information processing systems*, 37:110935–110971, 2024.

Xiaojiang Zhang et al. Srpo: A cross-domain implementation of large-scale reinforcement learning on llm. *arXiv preprint arXiv:2504.14286*, 2025.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. Galore: Memory-efficient llm training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*, 2024.

Hanqing Zhu, Zhenyu Zhang, Wenyan Cong, Xi Liu, Sem Park, Vikas Chandra, Bo Long, David Z Pan, Zhangyang Wang, and Jinwon Lee. Apollo: Sgd-like memory, adamw-level performance. *arXiv preprint arXiv:2412.05270*, 2024.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

# A Revisiting LoRA through the Lens of Our Theory

A recent report (Schulman & Lab, 2025) finds that low-rank LoRA, even rank-1, can match full-parameter RL performance. Our theory offers an explanation: in full-parameter RL, effective updates lie *off* the principal directions and induce only small spectral changes. Low-rank adapters can approximate these off-principal updates, while freezing the base weights regularizes training and discourages moves toward principal directions. With an appropriately scaled learning rate, the limited adapter capacity is therefore sufficient to catch up to full-parameter performance at least in short run.

**However**, the same report suggests principal-targeted variants such as **PiSSA** (Meng et al., 2024) should yield further gains. Our geometry account disagrees: aligning updates to top-$r$ principal directions enforces SFT-style behavior that is *misaligned* with RLVR's off-principal bias.

**Empirical test.** On `DS-Qwen-1.5B` with `DeepMath-103K` (He et al., 2025), we sweep ranks $\{8, 32, 64\}$ and learning rates $\{1\times10^{-4}, 5\times10^{-5}, 1\times10^{-5}\}$ for 200 steps, and report $\mathrm{pass@1}$ (mean over 16 samples) on AIME24 and AMC23 (Fig. 8). To control for model effects, we repeat on `Llama-3.2-3B-Instruct` with a Math corpus and report $\mathrm{pass@1}$ (mean over 4) on MATH500 (Fig. 9).

**Our findings.** Across settings, the principal-targeted *PiSSA* provides no clear gain over LoRA. At the higher learning rates used for low-rank adapters to match full-parameter performance, PiSSA *often becomes unstable and collapses* earlier than LoRA. This occurs because scaling the learning rate in PiSSA *enforces updates along principal directions*, higher-curvature and spectrum-distorting, precisely the directions RLVR tends to avoid. The result is brittle optimization and early collapse, whereas LoRA's off-principal updates remain better aligned with RLVR's geometry.

**Insight.** These results support the geometry-based account: principal-aligned LoRA variants are *over-fit to SFT's update geometry* and misaligned with RL's training dynamics, so success in SFT does not transfer to RL.

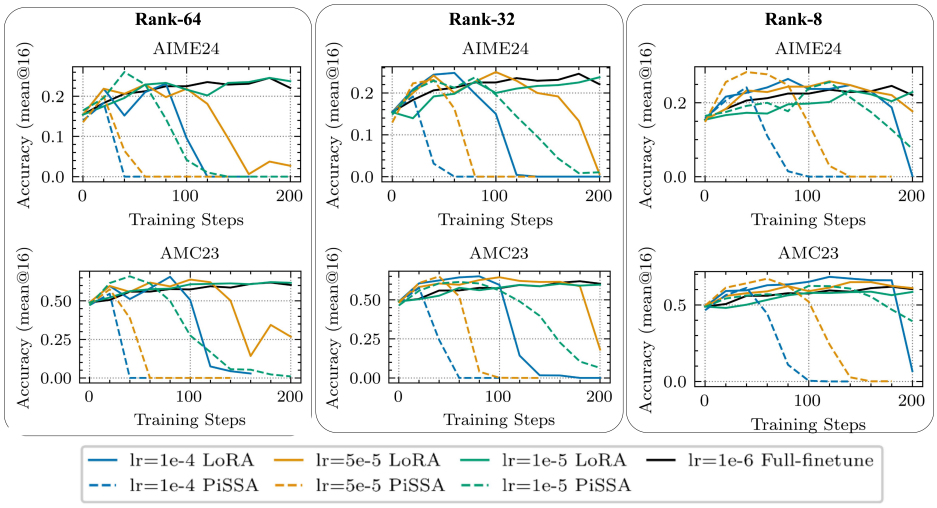


Figure 8: **LoRA vs. PiSSA on** `DS-Qwen-1.5B` **(DeepMath-103K).** We sweep ranks $\{8, 32, 64\}$ and learning rates $\{1\times10^{-4}, 5\times10^{-5}, 1\times10^{-5}\}$ for 200 steps, reporting pass@1 (avg@16) on AIME24 (top) and AMC23 (bottom). Across settings, **PiSSA** (principal-targeted) provides *no additional gains* over LoRA and, at higher learning rates that force principal-direction updates, *often collapses early*; LoRA remains more stable. This supports our geometric account: forcing updates into principal directions (favored in SFT) is misaligned with RL, offering no obvious gain and leading to training collapse when scaling up learning rates.

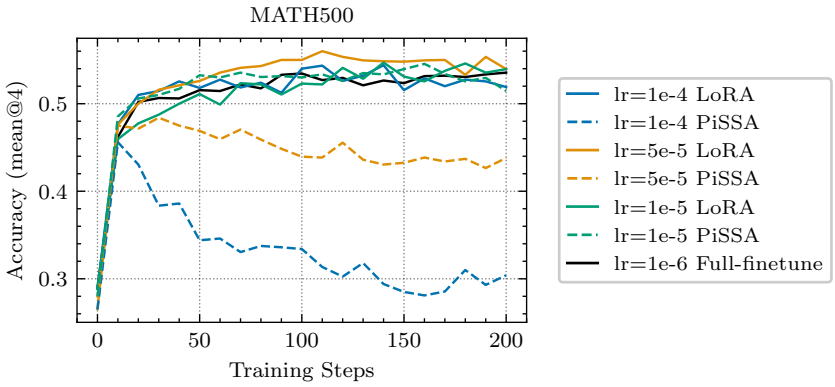


Figure 9: **LoRA vs. PiSSA on** `LLaMA-3.2-3B`**.** We sweep learning rates $\{1\times10^{-4}, 5\times10^{-5}, 1\times10^{-5}\}$ with a fixed rank of 64 for 200 steps, reporting pass@1 (mean@4) on MATH500. Consistent with the `DS-Qwen-1.5B` results in Fig. 8, **PiSSA** provides *no additional gain* over LoRA and, under higher learning rates that emphasize principal-direction updates, *often collapses early*.

## B Clarification of LLM Usage

In this work, we employ LLMs to polish the writing throughout the paper and to assist in generating code for figure plotting. Besides, we use it for drawing the teaser figure.

## C More Related works

**Post-training** Large-scale models pre-trained on broad domains serve as general-purpose backbones with extensive domain knowledge and notable zero-shot capabilities (Radford et al., 2021; Achiam et al., 2023; Touvron et al., 2023; Hu et al., 2023; Li et al., 2024; Radford et al., 2018; Brown et al., 2020). However, such pre-trained models often fail to meet the specific application requirements or align with domain-specific constraints. *Post-training* methods address this gap by adapting foundation models to downstream tasks. Common approaches include supervised fine-tuning on curated datasets (Howard & Ruder, 2018; Dodge et al., 2020; Wei et al., 2021; Chung et al., 2024), reinforcement learning from human or automated feedback (Ziegler et al., 2019; Ouyang et al., 2022; Guo et al., 2025; Zhai et al., 2024), and other recent techniques (Rafailov et al., 2023).

Especially, the recent advances in LLM reasoning (DeepSeek-AI, 2025) highlight the effectiveness of *Reinforcement Learning with Verifiable Rewards* (RLVR), which replaces subjective human judgments with automatically verifiable signals. RLVR has been shown to significantly enhance reasoning ability using policy optimization algorithms such as PPO (Ouyang et al., 2022) and GRPO (Shao et al., 2024). Building on these successes, a growing body of work (Yu et al., 2025; Liu et al., 2025b; Luo et al., 2025a; Zhang et al., 2025; Liu et al., 2025a; Xiong et al., 2025) continues to refine RL methods tailored for LLM reasoning.

**SFT versus RL.** Prior work comparing these paradigms has largely focused on downstream performance. A foundational result shows that on-policy RL can outperform offline SFT even with the same expert data (Ross et al., 2011). Recent empirical studies consistently reinforce this, finding that RL-tuned models often generalize better out-of-distribution (Han et al., 2025; Chu et al., 2025) and transfer more effectively to new tasks (Huan et al., 2025) than their SFT counterparts.

While these studies establish a performance hierarchy, our work investigates a different dimension: how these distinct methods affect the model's internal structure. A recent study observed that RL fine-tunes only a fraction of the network's parameters (Mukherjee et al., 2025), but this empirical finding left the underlying mechanism unexplored and did not characterize or predict the affected subnetwork. Our work aims to bridge this gap by providing a mechanistic explanation for this phenomenon.

## D Experimental Details

### D.1 Training Settings

**Models & Datasets.** We run post-training experiments on three open models: **DeepSeek-R1-Distill-Qwen-1.5B** (Yang et al., 2024), **Qwen2.5-Math-7B** (Yang et al., 2024), and **Qwen3-Base** (Team, 2025). The maximum context length is set to 8192 for DeepSeek-R1-Distill-Qwen-1.5B and Qwen2.5-Math-7B, and to 20480for Qwen3-4B-Base.

We evaluate primarily on mathematics using two training corpora to reduce dataset-specific confounds. (1) **DAPO+MATH (DM):** a union of the DAPO-Math-17k set[2] and the MATH dataset (Hendrycks et al., 2021). (2) **DS+SR:** the 47k DeepScaler collection (Luo et al., 2025b) combined with high-difficulty (levels 3–5) problems extracted from SimpleRL (Zeng et al., 2025a).We use the version from Huan et al. (2025).

**Training details.** We implement RLVR on the VeRL pipeline (Sheng et al., 2024) (v4.0) and use vLLM (Kwon et al., 2023)(v8.5) for rollouts. We use FSDPv2 with the default mixed precision configuration. All experiments run on NVIDIA H200 GPUs. Unless otherwise noted, we use DAPO (Yu et al., 2025) *without* an explicit reference-KL penalty (ratio clipping as in DAPO), a global batch

[2]DAPO-Math-17k

size of 256 (mini-batch 64) with 4 gradient update per step. We use **DAPO** primarily to eliminate the confounding effect of the KL penalty, which can otherwise obscure the intrinsic parameter update dynamics during training.

Per-model configurations without specific mention:

- **Qwen2.5-Math-7B** on **DM**: 16 rollouts per prompt; 8 x H200 GPUs; 300 training steps.
- **DeepSeek-R1-Distill-Qwen-1.5B** on **DS+SR**: 12 rollouts per prompt; 16 x H200 GPUs; 320 steps.
- **Qwen3-4B-Base** on **DS+SR**: 16 rollouts per prompt; 32 x H200 GPUs; 150 steps.

For LoRA and PiSSA studies, to reduce compute cost during learning-rate sweeps, we use the same DAPO recipe with a global batch size of 128 (mini-batch 32), four gradient updates per step, and 16 rollouts per prompt. Both **DeepSeek-R1-Distill-Qwen-1.5B** and **LLaMA-3.2-3B** are trained for 200 steps.

The actor is optimized with AdamW (Loshchilov & Hutter, 2017) (constant learning rate $1\times10^{-6}$, $\beta_1$=0.9, $\beta_2$=0.999). Rewards are *verifiable*: +1.0 if the extracted final answer is correct and −1.0 otherwise (no separate format score), following the verifier implementation of Su et al. (2025). We enable an over-length penalty with an additional 1024-token budget and a penalty factor of 1.0.

### D.2 Evaluation settings

We evaluate models on four widely used benchmarks: AIME24 (MAA, 2024), AIME25 (MAA, 2025), AMC23 (MAA, 2023), MATH-500 (Lightman et al., 2023), as we main train using math daastets. We used Eval-Chemy (Raoof et al., 2025) with their default temperature 0.7 and 0.8 as the top-p value. In our experiments, we used **the averaged accuracy**, i.e., $pass@1(avg@k)$ for all benchmarks. to evaluate the models' performance. Specifically, for AIME24 and AIME 25, we averaged accuracy on 64 samples, for AMC, we average accuracy on 32 samples, For MATH 500, our score is the average accuracy over 2 samples.

## E Intervention details

**Intervention 1: loss–preserving V/O rotation.** Let $D$ be the head dimension, $H_q$ the number of query heads, $H_{kv}$ the number of key/value heads, and $n_{\text{rep}} = H_q/H_{kv}$ (grouped GQA). Denote

$$W_v \in \mathbb{R}^{d_{\text{model}}\times(H_{kv}D)}, \qquad W_o \in \mathbb{R}^{d_{\text{model}}\times(H_qD)}.$$

Draw any orthogonal $R \in \mathbb{R}^{D\times D}$ (Haar/Hadamard) and form the block rotations

$$R_{kv} = \operatorname{diag}(\underbrace{R,\ldots,R}_{H_{kv}}) \in \mathbb{R}^{(H_{kv}D)\times(H_{kv}D)}, \qquad R_q = \operatorname{diag}(\underbrace{R,\ldots,R}_{n_{\text{rep}}},\underbrace{R,\ldots,R}_{n_{\text{rep}}},\ldots) \in \mathbb{R}^{(H_qD)\times(H_qD)}.$$

We edit the weights by right–multiplication along the head axis:

$$\boxed{W_v' = W_v R_{kv}, \qquad W_o' = W_o R_q.} \tag{11}$$

If $b_v$ exists, reshape $b_v$ per head and set $b_v' = b_v R_{kv}$.

**Proposition E.1** (Exact invariance)**.** *Let* $\mathrm{Ctx} = \mathrm{Attn}(Q,K,V) \in \mathbb{R}^{\cdot\times(H_qD)}$. *Under equation 11,*

$$\mathrm{out}' = \mathrm{Attn}(Q,K,VR_{kv})\,(W_oR_q)^\top = \mathrm{Ctx}\,R_qR_q^\top W_o^\top = \mathrm{Ctx}\,W_o^\top = \mathrm{out}.$$

**Intervention 2: head shuffle (lossless).** Let $P_{kv}$ be a permutation of the $H_{kv}$ KV heads and $P_q$ its grouped expansion to $H_q$ heads. Apply

$$\text{cols of } (W_k, W_v) \leftarrow P_{kv}, \qquad \text{cols of } W_q \leftarrow P_q, \qquad \text{columns of } W_o \leftarrow P_q^{-1}.$$

This relabels which head carries which subspace, while leaving the block function unchanged.

We show that after weight intervention, the model weights update position has a sub-random overlap while those untouched weights stay a high overlap.

# F Examples of why previous identified method fails

## F.1 Failures of a Fixed Absolute Tolerance Rule

- **False positives at large scale.** Within $[2^{10}, 2^{11})$=$[1024, 2048)$, the bf16 spacing is $\mathrm{ULP}_{\mathrm{bf16}} = 2^{10-7} = 8$. Numbers like $1024.001$ and $1024.002$ differ by $10^{-3}$>$10^{-5}$, hence would be flagged as "changed" by the $10^{-5}$ rule, yet both round to the same bf16 code (1024), i.e., *no storage-level change*.
- **False negatives at small scale.** Around $10^{-6} \approx 2^{-20}$, the bf16 spacing is $\mathrm{ULP}_{\mathrm{bf16}} = 2^{-27} \approx 7.45\times10^{-9}$. Weights $w$=$10^{-6}$ and $\widehat{w}$=$2\times10^{-6}$ differ by $10^{-6}$≤$10^{-5}$ and would be marked "equal" by the $10^{-5}$ rule, yet they are separated by ≈ 134 ULPs and quantize to *different* bf16 codes.

## F.2 Justification of our probe

**Lemma F.1** (Gap between distinct bf16 representables)**.** *If $x \neq y$ are normalized bf16 numbers in the same binade $[2^e, 2^{e+1})$, then*

$$|x - y| \;\geq\; 2^{\,e-7} \quad \textit{and} \quad \frac{|x-y|}{\max(|x|,|y|)} \;>\; 2^{-8}.$$

*The strict inequality also holds across the binade boundary.*

**Lemma F.2** (ULP lens: magnitude-dependent threshold)**.** *For normalized bf16 values $x$ with $|x| \in [2^e, 2^{e+1})$,*

$$\frac{\mathrm{ULP}_{\mathrm{bf16}}(x)}{|x|} \in (2^{-8},\, 2^{-7}] \;=\; (0.390625\%,\, 0.78125\%].$$

*Hence the* minimal realized *relative update at magnitude $|x|$ is $\gtrsim \frac{1}{2}\,\mathrm{ULP}_{\mathrm{bf16}}(x)/|x| \in (0.195\%,\, 0.391\%]$. In particular,* larger $|x|$ *requires a* larger *absolute step to register.* □

**Proposition F.3** (Soundness and completeness of the probe)**.** *Let $w_i, \widehat{w}_i$ be* normalized *bf16 values (finite, nonzero), and suppose $\eta < \frac{1}{2}\min_x \mathrm{ULP}_{\mathrm{bf16}}(x)/|x| = 2^{-9} \approx 1.953 \cdot 10^{-3}$. Then*

$$\left|\,\widehat{w}_i - w_i\,\right| \leq \eta \max(|w_i|, |\widehat{w}_i|) \quad \Longleftrightarrow \quad \mathrm{bf16}(w_i) = \mathrm{bf16}(\widehat{w}_i).$$

*Proof.* (⇒)If $w_i \neq \widehat{w}_i$, Lemma F.2 gives $|\widehat{w}_i - w_i| / \max(|w_i|, |\widehat{w}_i|) > 2^{-8} > 2\eta$, contradiction. Hence $w_i = \widehat{w}_i$ as bf16 numbers.

(⇐) If the stored bf16 values are equal, the difference is 0, which satisfies equation 1. □ □

**Corollary F.4** (Choice $\eta = 10^{-3}$ is safe)**.** *Since $10^{-3} < 2^{-9}$, Proposition F.3 applies: the test equation 1 passes* iff *the two bf16 entries are bit-wise identical (or both zero). Thus $\eta = 10^{-3}$ yields a* scale-aware *probe that flags equality only when storage is unchanged.*

# G Math Analysis

## G.1 Policy-Gradient Fine-Tuning (DAPO)

Assume an *old* policy $\pi_{\text{old}}$ that we use to sample $G$ candidate completions $y^{1:G}$ for each prompt $x \in \mathcal{X}$. For a single token $y_{i,t}$ (token $t$ in completion $i$) we define the *importance-weighted advantage*

$$w_{i,t} \;=\; \underbrace{\frac{\pi_\theta(y_{i,t}|x, y_{<t})}{\pi_{\text{old}}(y_{i,t}|x, y_{<t})}}_{\text{importance ratio}}\, \hat{A}_{i,t}\, \mathbb{I}_{\text{clip}} \quad \in \mathbb{R}, \tag{1}$$

where $\hat{A}_{i,t}$ is the estimated advantage and $\mathbb{I}_{\text{clip}} \in \{0, 1\}$ implements the usual trust-region clipping.

**Token-level objective.** The DAPO loss can be written as a sum of weighted log-probabilities

$$J_{\text{RL}}(\theta) \;=\; \mathbb{E}_{x\sim\mathcal{X},\, y^{1:G}\sim\pi_{\text{old}}}\Big[\tfrac{1}{\sum_i |y^i|}\sum_{i=1}^{G}\sum_{t=1}^{|y^i|} w_{i,t}\, \log \pi_\theta(y_{i,t} \mid x, y^i_{<t})\Big]. \tag{2}$$

### G.2 Proof of Gate I: On-Policy RL Implies a One-Step KL Leash

This appendix provides the standard tilting oracle and $M$-projection facts, local second-order expansions, and the proof of the one-step policy-KL leash (Prop. 3.1 in the main text). ***We keep the proof concise, otherwise too lengthy, especially for those has shown in some prior work Shenfeld et al. (2025); Wu et al. (2025).*** Our one-step analysis is inspired by recent work Wu et al. (2025); Shenfeld et al. (2025), which uses a similar variational approach to show that even the final converged policy remains KL-proximal to the base policy. We also record a trust-region/clipping bound used when $\beta = 0$.

Throughout, $x$ is fixed, $q(\cdot \mid x)$ has full support on $\mathcal{Y}$, and $\pi_\theta(\cdot \mid x)$ is a $C^3$ parametric family with log-density $\log \pi_\theta$ locally smooth. Expectations without explicit subscript are conditional on $x$.

We first show useful lemmas here.

**Lemma G.1** (Frozen-policy surrogate is second-order tight)**.** *Let $f(\theta) := \mathcal{L}_{\mathrm{PG}}(\theta)$ in equation 4 and $g(\theta) := \widetilde{\mathcal{L}}_{\mathrm{PG}}(\theta; \theta_t)$ be the frozen-policy surrogate with $A_{\theta_t}$. Then $f(\theta_t) = g(\theta_t)$ and $\nabla f(\theta_t) = \nabla g(\theta_t)$. If $\nabla f$ and $\nabla g$ are $L$-Lipschitz in a neighborhood of $\theta_t$, then*

$$| f(\theta_t + \Delta\theta) - g(\theta_t + \Delta\theta) | \;\le\; \tfrac{L}{2} \, \|\Delta\theta\|^2.$$

Proof. *At $\theta_t$, both objectives evaluate to $-\mathbb{E}_{\pi_{\theta_t}}[A_{\theta_t} \log \pi_{\theta_t}]$. For the gradient, using the log-derivative trick and the centering of $A_{\theta_t}$, both yield $-\mathbb{E}_{\pi_{\theta_t}}[A_{\theta_t} \nabla \log \pi_{\theta_t}]$. Thus $f(\theta_t) = g(\theta_t)$ and $\nabla f(\theta_t) = \nabla g(\theta_t)$. The bound is the standard second-order Taylor remainder under Lipschitz gradients.* □

**1: Exponential tilting and M-projection**

**Lemma G.2** (Gibbs variational principle / exponential tilting)**.** *Fix $\beta > 0$ and a full-support reference $q(\cdot \mid x)$. Then*

$$\max_{\pi \ll q} \; \Big\{ \mathbb{E}_{y\sim\pi}[R(x,y)] - \beta \, D_{\mathrm{KL}}(\pi \| q) \Big\}$$

*is uniquely maximized by*

$$\tilde{q}_\beta(y \mid x) = \frac{q(y \mid x) \exp(R(x,y)/\beta)}{\mathbb{E}_{y\sim q}[\exp(R(x,y)/\beta)]}.$$

*Proof.* Consider $\mathcal{L}(\pi, \lambda) = \mathbb{E}_\pi[R] - \beta \mathbb{E}_\pi\big[\log \frac{\pi}{q}\big] + \lambda(\sum_y \pi(y) - 1)$. Stationarity in $\pi$ gives $\log \frac{\pi}{q} = R/\beta - \lambda - 1$, hence $\pi \propto q \, e^{R/\beta}$. Strict concavity in $\pi$ yields uniqueness. □

**Lemma G.3** (Policy Gradient Update as Parametric $M$-projection)**.** *For fixed $\tilde{q}_\beta$,*

$$\arg\min_\theta \; D_{\mathrm{KL}}(\tilde{q}_\beta \| \pi_\theta) \;=\; \arg\max_\theta \; \mathbb{E}_{y\sim\tilde{q}_\beta}[\log \pi_\theta(y \mid x)].$$

*Proof.* $D_{\mathrm{KL}}(\tilde{q}_\beta \| \pi_\theta) = \mathbb{E}_{\tilde{q}_\beta}[\log \tilde{q}_\beta] - \mathbb{E}_{\tilde{q}_\beta}[\log \pi_\theta]$, where the first term is $\theta$-independent. We omit the full proof here, with one can be found in Shenfeld et al. (2025). □

**2: Local second-order identities**

**Lemma G.4** (Local Pythagorean identity for the $M$-projection)**.** *Let $f(\theta) := D_{\mathrm{KL}}(\tilde{q}_\beta \| \pi_\theta) = \mathbb{E}_{\tilde{q}_\beta}[-\log \pi_\theta] + \mathrm{const}$. Assume $\log \pi_\theta$ is $C^3$ near $\theta$, and let $\theta^+ \in \arg\min f$. Writing $\Delta := \theta^+ - \theta$, for $\|\Delta\|$ small,*

$$f(\theta) - f(\theta^+) = \tfrac{1}{2} \, \Delta^\top H_{\tilde{q}}(\theta) \, \Delta + O(\|\Delta\|^3), \quad H_{\tilde{q}}(\theta) := -\mathbb{E}_{\tilde{q}_\beta}[\nabla^2 \log \pi_\theta].$$

*Proof.* Taylor-expand $f$ at $\theta^+$: $f(\theta) = f(\theta^+) + \frac{1}{2}\Delta^\top H_{\tilde{q}}(\theta^+)\Delta + O(\|\Delta\|^3)$ since $\nabla f(\theta^+) = 0$. Local $C^3$ smoothness implies $H_{\tilde{q}}(\theta^+) = H_{\tilde{q}}(\theta) + O(\|\Delta\|)$, which is absorbed into the cubic remainder. □

**Lemma G.5** (Quadratic expansion of policy KL)**.** *Let $F(\theta) := -\mathbb{E}_{\pi_\theta}[\nabla^2 \log \pi_\theta]$ be the Fisher information. Then*

$$D_{\mathrm{KL}}(\pi_{\theta+\Delta} \| \pi_\theta) = \tfrac{1}{2} \, \Delta^\top F(\theta) \, \Delta + O(\|\Delta\|^3).$$

*Proof.* Expand $\log \frac{\pi_{\theta+\Delta}}{\pi_\theta} = \Delta^\top \nabla \log \pi_\theta + \frac{1}{2}\Delta^\top \nabla^2 \log \pi_\theta \, \Delta + O(\|\Delta\|^3)$, take expectation under $\pi_{\theta+\Delta} = \pi_\theta + O(\|\Delta\|)$, use $\mathbb{E}_{\pi_\theta}[\nabla \log \pi_\theta] = 0$ and $-\mathbb{E}_{\pi_\theta}[\nabla^2 \log \pi_\theta] = F(\theta)$. □

**3. Relating projection Hessian and Fisher under small tilt**

**Lemma G.6** (Hessian–Fisher proximity)**.** *Suppose* $\|\nabla^2 \log \pi_\theta(y \mid x)\|_{\mathrm{op}} \le L$ *uniformly near* $\theta$*. Then*

$$\left\|H_{\tilde{q}}(\theta) - F(\theta)\right\|_{\mathrm{op}} \;\le\; 2L\,\mathrm{TV}(\tilde{q}_\beta, \pi_\theta) \;\le\; L\sqrt{2\,D_{\mathrm{KL}}(\tilde{q}_\beta \| \pi_\theta)}.$$

*In particular, with* $\kappa := D_{\mathrm{KL}}(\tilde{q}_\beta \| \pi_\theta) \to 0$*, we have* $H_{\tilde{q}}(\theta) = (1 + O(\sqrt{\kappa}))\, F(\theta)$ *as quadratic forms.*

*Proof.* For bounded matrix-valued $h$, $\|\mathbb{E}_{\tilde{q}} h - \mathbb{E}_\pi h\|_{\mathrm{op}} \le 2\|h\|_\infty \,\mathrm{TV}(\tilde{q}, \pi)$. Apply this with $h := -\nabla^2 \log \pi_\theta$ and Pinsker's inequality $\mathrm{TV}(p, q) \le \sqrt{\frac{1}{2} D_{\mathrm{KL}}(p \| q)}$. □

**4. Remainder control**

**Lemma G.7** (Cubic remainder is $o(f)$)**.** *If* $H_{\tilde{q}}(\theta) \succeq mI$ *on the update subspace (local strong convexity), then for* $\|\Delta\|$ *small*

$$\|\Delta\|^2 \;\le\; \tfrac{2}{m}\left(f(\theta) - f(\theta^+)\right), \qquad O(\|\Delta\|^3) = o\big(f(\theta)\big).$$

*Proof.* From Lemma G.4, $f(\theta) - f(\theta^+) \ge \frac{m}{2}\|\Delta\|^2 + O(\|\Delta\|^3)$. Rearranging yields $\|\Delta\|^2 = O(f(\theta) - f(\theta^+))$, so the cubic term is lower order. □

### G.2.1 Proof of Proposition 3.1

*Proof of Proposition 3.1.* Let $f(\theta) = D_{\mathrm{KL}}(\tilde{q}_\beta \| \pi_\theta)$ and $\Delta = \theta^+ - \theta$. By Lemma G.4,

$$f(\theta) - f(\theta^+) = \tfrac{1}{2}\,\Delta^\top H_{\tilde{q}}(\theta)\Delta + O(\|\Delta\|^3).$$

By Lemma G.5,

$$D_{\mathrm{KL}}(\pi_{\theta^+} \| \pi_\theta) = \tfrac{1}{2}\,\Delta^\top F(\theta)\Delta + O(\|\Delta\|^3).$$

By Lemma G.6 with $\kappa = f(\theta)$, $\Delta^\top F \Delta = (1 + O(\sqrt{\kappa}))\,\Delta^\top H_{\tilde{q}} \Delta$. Hence

$$D_{\mathrm{KL}}(\pi_{\theta^+} \| \pi_\theta) = (1 + O(\sqrt{\kappa}))\left(f(\theta) - f(\theta^+)\right) \;+\; O(\|\Delta\|^3).$$

Since $f(\theta^+) \ge 0$, $f(\theta) - f(\theta^+) \le f(\theta) = \kappa$. By Lemma G.7, $O(\|\Delta\|^3) = o(f(\theta))$. Therefore

$$D_{\mathrm{KL}}(\pi_{\theta^+} \| \pi_\theta) \;\le\; (1 + o(1))\, f(\theta) \;=\; (1 + o(1))\, D_{\mathrm{KL}}(\tilde{q}_\beta \| \pi_\theta),$$

which is the desired inequality. □

### G.2.2 Proof of Proposition 3.2

*Proof of Proposition 3.2.* By the quadratic expansion of policy KL (Lemma G.5),

$$D_{\mathrm{KL}}(\pi_{\theta+\Delta} \| \pi_\theta) = \tfrac{1}{2}\,\Delta^\top F(\theta)\Delta \;+\; R(\Delta), \qquad |R(\Delta)| \;\le\; C\,\|\Delta\|^3 \tag{12}$$

for some local constant $C > 0$ (from $C^3$ smoothness). Let $a := \Delta^\top F(\theta)\Delta$. Using the spectral lower bound $F(\theta) \succeq \mu I$ on the update subspace,

$$\|\Delta\|^2 \;\le\; \tfrac{a}{\mu}. \tag{13}$$

Combining equation 12–equation 13 yields

$$D_{\mathrm{KL}}(\pi_{\theta+\Delta} \| \pi_\theta) \;\ge\; \tfrac{1}{2}\,a \;-\; C\left(\tfrac{a}{\mu}\right)^{3/2}.$$

Since $D_{\mathrm{KL}}(\pi_{\theta^+} \| \pi_\theta) \le K$, we have

$$K \;\ge\; \tfrac{1}{2}\,a \;-\; C\,\mu^{-3/2} a^{3/2}. \tag{14}$$

For $a$ sufficiently small (equivalently, $K$ small), the cubic term is dominated by the linear term: choose $a_0 > 0$ so that $C\,\mu^{-3/2}\sqrt{a} \le \frac{1}{4}$ whenever $0 < a \le a_0$. Then from equation 14

$$K \;\ge\; \left(\tfrac{1}{2} - \tfrac{1}{4}\right)a \;=\; \tfrac{1}{4}\,a \quad\Rightarrow\quad a \;\le\; 4K.$$

Substituting $a \le 4K$ back into equation 12 refines the remainder: $|R(\Delta)| \le C\|\Delta\|^3 \le C(a/\mu)^{3/2} = O(K^{3/2}) = o(K)$, so $D_{\mathrm{KL}}(\pi_{\theta+\Delta}\|\pi_\theta) = \frac{1}{2}a + o(K)$. Hence $a = 2\,D_{\mathrm{KL}}(\pi_{\theta+\Delta}\|\pi_\theta) + o(K) \le 2K + o(K)$, i.e.

$$\Delta^\top F(\theta)\Delta \;\le\; 2K\,(1+o(1)).$$

Taking square roots gives the Fisher-norm bound in equation 6: $\|\Delta\|_{F(\theta)} = \sqrt{\Delta^\top F(\theta)\Delta} \le \sqrt{2K}\,(1+o(1))$. The Euclidean bound follows from equation 13:

$$\|\Delta\|_2 \;\le\; \sqrt{\tfrac{\Delta^\top F(\theta)\Delta}{\mu}} \;\le\; \sqrt{\tfrac{2K}{\mu}}\,(1+o(1)).$$

Finally, for any parameter block $W \subset \theta$, its Frobenius change is the $\ell_2$-norm of the corresponding subvector of $\Delta$; therefore $\|\Delta W\|_F \le \|\Delta\|_2$. □

#### G.2.3 One-step KL budget (used in Gate II)

**Corollary G.8** (KL budget). *If $D_{\mathrm{KL}}(\pi_{\theta^+}\|\pi_\theta) \le K$, then*

$$\tfrac{1}{2}\,\Delta^\top F(\theta)\,\Delta \;\le\; K\,(1+o(1)).$$

*Proof.* Apply Lemma G.5 and Lemma G.7. □

#### G.2.4 Trust-region / clipping bound (for $\beta = 0$)

**Lemma G.9** (Implicit KL leash from ratio clipping). *Let $r_t = \frac{\pi_{\theta^+}(y_t|x,y_{<t})}{\pi_\theta(y_t|x,y_{<t})}$ and suppose clipping enforces $r_t \in [1-\varepsilon,\,1+\varepsilon]$ on the batch. Then*

$$\widehat{D}_{\mathrm{KL}}(\pi_{\theta^+}\|\pi_\theta) \;\le\; \widehat{\mathbb{E}}[T(x)]\cdot\max\{-\log(1-\varepsilon),\log(1+\varepsilon)\} \;=\; O(\varepsilon)\cdot\widehat{\mathbb{E}}[T(x)],$$

*and in the small-step regime (mean-zero advantage) this tightens to $O(\varepsilon^2)$.*

*Proof.* Autoregressive factorization gives $D_{\mathrm{KL}}(\pi_{\theta^+}\|\pi_\theta) = \mathbb{E}_{\pi_{\theta^+}}[\sum_t \log r_t]$. Because $\log r_t \in [\log(1-\varepsilon),\log(1+\varepsilon)]$, we have $|\log r_t| \le c(\varepsilon)$; summing over $t$ and taking batch expectation yields the stated bound. Using $\log(1\pm\varepsilon) = \pm\varepsilon + O(\varepsilon^2)$ and small-step arguments gives $O(\varepsilon^2)$. □

### G.3 Proofs for Gate II (Sec. 3.2)

**Setup (layer-conditioned budget).** Partition $\theta = (\mathrm{vec}(W), \theta_{\neg W})$ and let the Fisher at $\theta = \theta_t$ be

$$F(\theta) = \begin{bmatrix} F_{W,W} & F_{W,\neg W} \\ F_{\neg W,W} & F_{\neg W,\neg W} \end{bmatrix} \succeq 0.$$

For a one-step update $\Delta\theta$, the global KL leash implies $\frac{1}{2}\Delta\theta^\top F(\theta)\Delta\theta \le K$. Define the layer-conditioned curvature

$$S_W := F_{W,W} - F_{W,\neg W}F_{\neg W,\neg W}^{-1}F_{\neg W,W} \succeq 0,$$

and the per-layer budget $\delta_W := \frac{1}{2}\,\mathrm{vec}(\Delta W)^\top S_W\,\mathrm{vec}(\Delta W) \le K$. Let $\mu_W := \lambda_{\min}(S_W) > 0$ on the update subspace.

**Lemma G.10** (Layer-conditioned Frobenius/operator bounds). *$\|\Delta W\|_F \le \sqrt{2\delta_W/\mu_W}$ and $\|\Delta W\|_2 \le \|\Delta W\|_F$.*

*Proof.* Since $S_W \succeq \mu_W I$, $\delta_W \ge \frac{1}{2}\mu_W\|\Delta W\|_F^2$. □ □

**Lemma G.11** (Wedin's sin–Θ). *For $W_+ = W_0 + \Delta W$, the principal subspace angles satisfy $\|\sin\Theta(U_k(W_0),U_k(W_+))\|_2 \le \|\Delta W\|_2/\gamma_k$ and similarly for $V_k$.* □

**Lemma G.12** (Weyl/Mirsky and Hoffman–Wielandt). *$|\sigma_k(W_+) - \sigma_k(W_0)| \le \|\Delta W\|_2$ and $\sum_i(\sigma_i(W_+)-\sigma_i(W_0))^2 \le \|\Delta W\|_F^2$.* □

**Corollary G.13** (Projection stability). *With the same assumptions,*

$$\big\|U_k(W_0)U_k(W_0)^\top - U_k(W_+)U_k(W_+)^\top\big\|_2 \;=\; \big\|\sin\Theta\big(U_k(W_0),U_k(W_+)\big)\big\|_2 \;\le\; \frac{\sqrt{2\delta_W/\mu_W}}{\gamma_k}.$$

*The analogous bound holds for the right subspaces with $V_k$.* Interpretation. *The leading invariant subspaces rotate by at most $O\big(\sqrt{\delta_W/\mu_W}/\gamma_k\big)$; when the gap is moderate, the rotation is small.* □

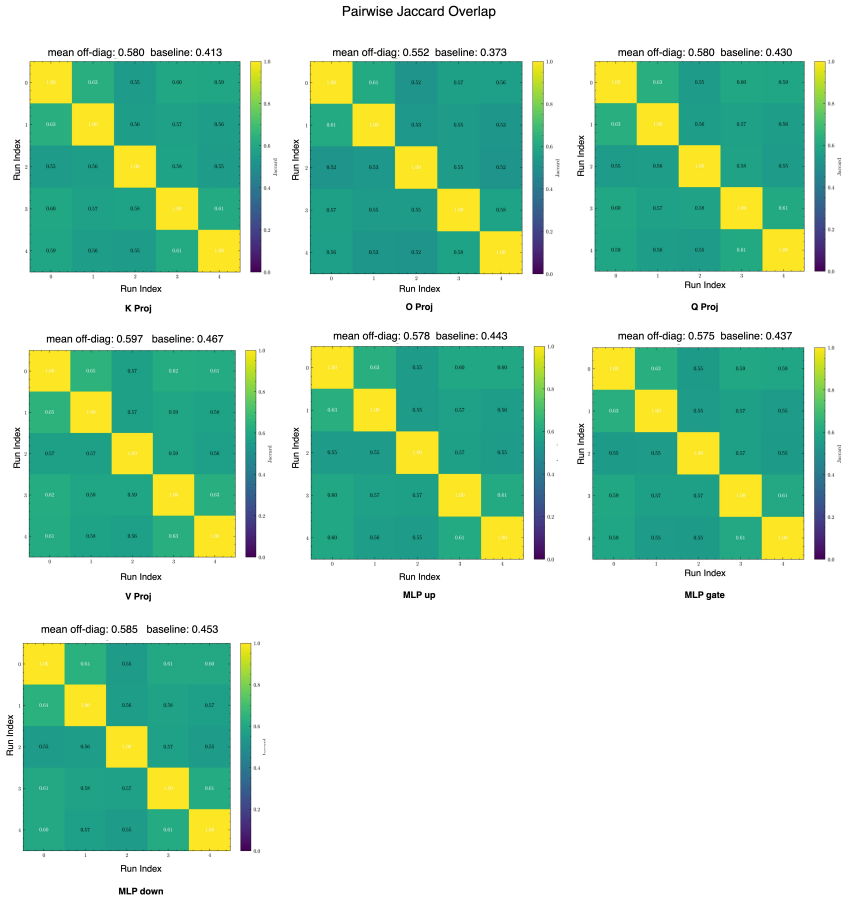


Figure 10: Pair-wise Jaccard similarity of update masks from five independent RLVR runs on Layer 13 of the `DS-Distill-Qwen-1.5B` model.

# H More Visualization

## H.1 Jaccard matrix

RL updates are highly consistent across independent training runs. Fig. 10 shows the pair-wise Jaccard similarity between the final update masks from five RLVR runs on different data and algorithms. The high similarity scores demonstrate that the optimization process consistently targets the same subset of parameters, providing strong evidence for a deterministic, non-random optimization bias.

## H.2 Spectrum shift for DS-1.5B and Qwen3-1

We also show the spectrum shift for DS-1.5B and Qwen3-14B here.

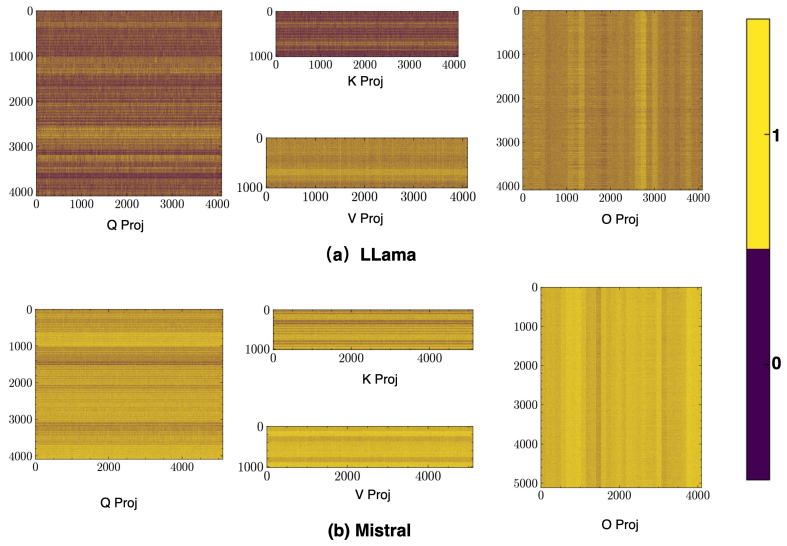


Figure 11: Structured Update observed on Llama(Llama-3.1-8B) and Mistral (Mistral-Small-24B) models. Here we plot the weight update mask using the zero-RL checkpoints from Zeng et al. (2025b).

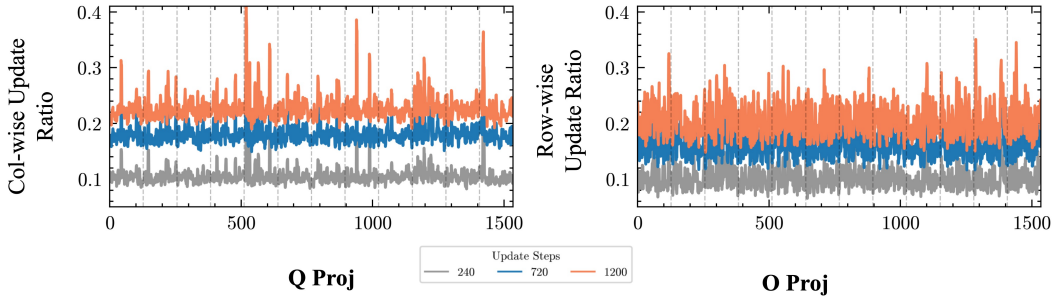


Figure 12: Temporal emergence of the optimization bias with row and column-wise update ratios for the 13th attention block across gradient update steps ($t \in \{240, 720, 1200\}$), smoothed with a 3-step window. *The column-wise (Q) and row-wise (O) update ratios show a much weaker bias.*

Table 3: Performance of DS-Qwen-1.5B with different masking strategies at 320 steps. Parameter counts shown are for linear layers only, excluding the embedding and head layers. Detailed evaluation settings are available in Appendix D.2. ***We observe that training only on principal weights $M_{princ}$ results in a clear accuracy gap compared to both the dense baseline and its complement $M^c_{princ}$. The models using the $M_{low}$ and $M^c_{princ} \cup M_{lowest}$ masks achieve performance closest to the dense baseline.***

| Model | Mask | Math500 | AMC23 | AIME24 | AIME25 | Average | #params |
|---|---|---|---|---|---|---|---|
| DS-Qwen-1.5B | Dense | 84.20 | **81.56** | **36.98** | **27.03** | **57.44** | 100% |
| | $M_{princ}$ | 83.60 | 77.19 | 30.16 | 24.32 | 53.82 | 50% |
| | $M^c_{princ}$ | 82.70 | 78.90 | 34.28 | 25.73 | 55.40 | 50% |
| | $M_{low}$ | 84.50 | 80.08 | 35.62 | 26.56 | 56.69 | 58.59% |
| | $M^c_{princ} \cup M_{low}$ | **85.20** | 78.83 | 34.74 | 26.20 | 56.24 | 74.02% |
| | Random-$M^c_{princ} \cup M_{low}$ | 84.50 | 77.35 | 34.48 | 25.01 | 55.34 | 74.02% |

Table 4: Performance of DS-Qwen-1.5B with different masking strategies with a extended training window to 500 steps. Parameter counts shown are for linear layers only, excluding the embedding and head layers. Detailed evaluation settings are available in Appendix D.2. ***We observe that training only on principal weights $M_{princ}$ results in a clear accuracy gap compared to both the dense baseline and its complement $M^c_{princ}$. The models using the $M_{low}$ and $M^c_{princ} \cup M_{lowest}$ masks achieve performance closest to the dense baseline.***

| Model | Mask | Math500 | AMC23 | AIME24 | AIME25 | Average | #params |
|---|---|---|---|---|---|---|---|
| DS-Qwen-1.5B | Dense | **84.5** | **83.52** | 38.28 | **28.075** | **58.59** | 100% |
| | $M_{princ}$ | 83.60 | 78.83 | 34.06 | 25.63 | 55.44 | 50% |
| | $M^c_{princ}$ | 84.0 | 77.97 | 38.64 | 27.81 | 56.90 | 50% |
| | $M_{low}$ | 83.8 | 82.42 | 37.03 | 27.82 | 57.77 | 58.59% |
| | $M^c_{princ} \cup M_{low}$ | 84.10 | 81.41 | **40.30** | 27.70 | 58.37 | 74.02% |
| | Random-$M^c_{princ} \cup M_{low}$ | 84.10 | 81.72 | 34.69 | 27.34 | 56.89 | 74.02% |

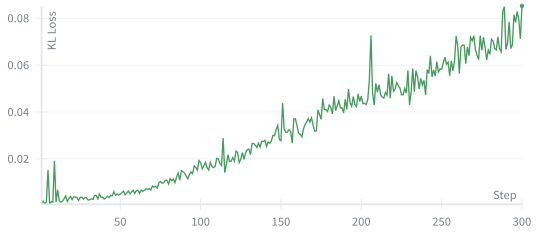


Figure 13: Token-wise KL loss. We show the token-wise KL loss during a DAPO run without a KL loss penalty, which shows a steadily increasing KL loss instead of being unconstrained.

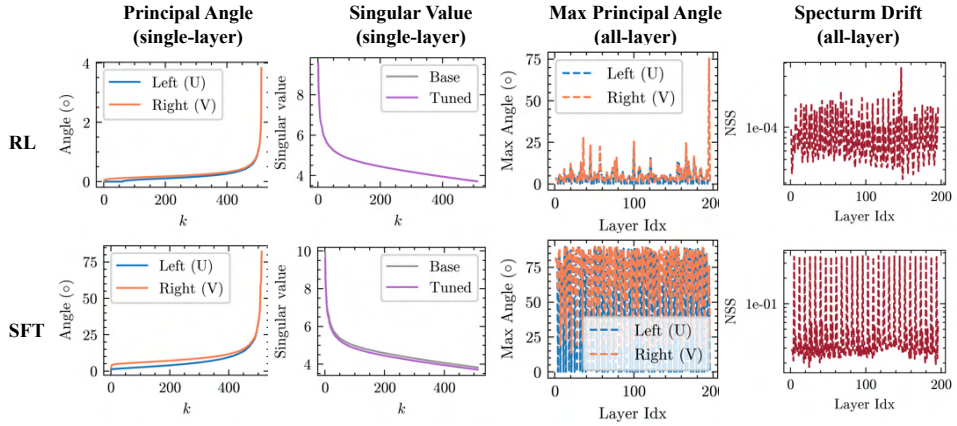


Figure 14: The spectrum probe results on the RL and SFT version on the `DS-Distill-Qwen-1.5B` Liu et al. (2025a). RLVR shows surprisingly stable top-k spectrum with minimal subspace rotation and top-k eigenvalue changes.

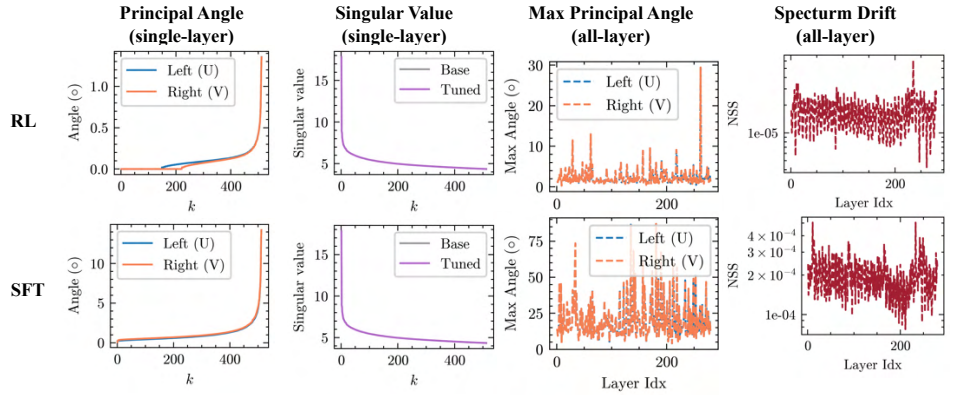


Figure 15: The spectrum probe results on the RL and SFT version on the Qwen3-14B Huan et al. (2025). RLVR shows surprisingly stable top-k spectrum with minimal subspace rotation and top-k eigenvalue changes.