The Path Not Taken: RLVR 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**[3,‡] **Hamed Pirsiavash**[1] **Jinwon Lee**[1] **David Z. Pan**[2]
Zhangyang Wang[2,†] **Yuandong Tian**[4,†,‡] **Kai Sheng Tai**[1,†,‡]
[1]Meta AI [2]The University of Texas at Austin [3]NVIDIA [4]Recursive Superintelligence

Abstract

RLVR reliably improves reasoning, yet it appears to update only a tiny fraction of weights. We resolve this paradox by revealing *a persistent, model-conditioned optimization bias*: independent runs concentrate updates in similar parameter regions, invariant to dataset or RL recipe, while finite-precision storage (e.g., bf16) obscures widespread micro-updates as a visual sparsity artifact. To characterize this unique bias, we provide a theoretical bridge: that *RLVR preferentially learns along off-principal directions* via a cascaded *Three-Gate mechanism*: updates are first bounded by an empirical KL constraint (Gate I), then steered by the model's anisotropic geometry (Gate II) toward spectrum-preserving off-principal subspaces, and finally filtered by finite precision (Gate III), effectively masking minor updates. Empirically, we validate the off-principal dynamics: RLVR exhibits minimal spectral drift, reduced principal-subspace rotation, and strong off-principal alignment that set RLVR strikingly apart from SFT. Together, our results provide the first parameter-space account of RLVR and uncover consistent regularities in weight evolution, advancing a more *white-box* understanding of RLVR. Moreover, we show that RLVR follows an optimization regime distinct from SFT, showing directly that transferring SFT-era PEFT can be flawed and motivating *geometry-aware, RLVR-native* methods.

1 Introduction

Large Reasoning Models (LRMs), such as OpenAI-o1 [15] and DeepSeek-R1 [6], have substantially improved their ability 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.

Sparsity paradox. Despite these advances, the parameter-level mechanisms by which RLVR reshapes a model remain poorly understood. Given the substantial computational resources devoted to RL [53] and its striking new behavioral divergence from supervised fine-tuning (SFT), one might assume significant weight changes. Surprisingly, recent evidence suggests the opposite: RLVR produces *apparently sparse* weight updates, whereas SFT updates are *dense* [35]. This creates a paradox: *a high-cost, high-gain procedure that seems to modify only a tiny fraction of weights.*

Optimization bias. We resolve this paradox by uncovering a deeper mechanism behind the apparent sparsity: a **persistent, model-conditioned optimization bias**. For a fixed pretrained model, RLVR concentrates visible updates into a narrow, stable subset of parameters, strikingly independent of runs, algorithms, or datasets. `bfloat16` precision then acts as a filter, obscuring the widespread micro-updates in non-preferred regions. As shown in Fig. 1, we depict this bias as an *implicit compass*: RLVR is subtly guided during optimization, even without the explicit teacher in SFT.

Code is available at this link. ‡Work done while at Meta AI. †Equal advisory contribution. ✉hqzhu@utexas.edu

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

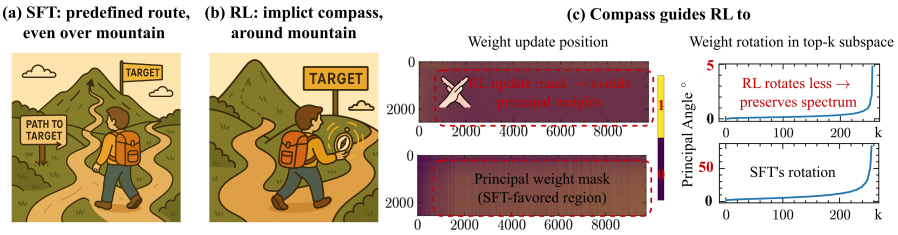

Figure 1: **SFT vs. RLVR: Optimization geometry.** (a) **SFT** follows external guidance, traversing high-curvature directions ("over the mountain"). (b) **RLVR** acts as if steered by an *implicit compass* (intrinsic model geometry), taking a low-curvature detour. (c) **Evidence.** *Left:* Positional maps comparing *update masks* with *principal masks* (high-energy weights in rank-k SVD [26]). RLVR updates distinctively avoid principal weights, whereas SFT targets them. *Right:* Principal angles show RLVR induces minimal subspace rotation (preserving spectral structure), while SFT distorts it significantly.

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 during training?

Three-Gate theoretical bridge. We explain RLVR's optimization dynamics through a Three-Gate Bridge. **Gate I (KL Anchor)** keeps the ideal projected improvement target close to the current policy, a KL-proximal regime that is empirically matched by on-policy RLVR. **Gate II (Model Geometry)** pinpoints the preference *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 makes the bias *model-conditioned*: it is inherited from the pretrained landscape, not imposed by a particular dataset or RL recipe. **Gate III (Precision)** then acts as a realization filter, masking sub-ULP micro-updates in non-preferred regions and making the off-principal bias *appear* sparse.

Experimental validation. We validate this bridge with a comprehensive suite of experiments, confirming that **RLVR learns off the principal directions**, operating in an optimization regime entirely distinct from SFT. We show that (i) RLVR *preserves* the pretrained spectral structure, 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 [26]; 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.

Guidance for learning algorithms. Consequently, directly carrying over SFT-era parameter-efficient fine-tuning (PEFT) methods, can be flawed, especially those over-aligned with SFT's optimization. *(1) Sparse fine-tuning.* Restricting updates to *principal* weights, an SFT prior [26], yields the weakest optimization trajectory, as reflected by KL drift [44] and accuracy. Conversely, updating *non-principal, low-magnitude* weights, precisely the off-principal regime (as predicted by our theory) closely tracks the dense RLVR trajectory. *(2) LoRA variants.* A recent report [42] finds that low-rank LoRA can match full-parameter RL performance. We test whether principal-component initialization, as used by PiSSA [33], provides further gains under RLVR. In our tested settings, this initialization provides no consistent advantage over LoRA and can be unstable at high learning rates.

Contributions. We make the following contributions:

- **Observation (Sec. 2).** We identify a *persistent, model-conditioned optimization bias* governing RLVR's parameter evolution, largely invariant to data and algorithms.
- **Theory (Sec. 3).** We propose the *Three-Gate Theory*: a theoretical bridge linking KL anchoring, model geometry, and finite-precision visibility to explain RLVR's off-principal update dynamics.
- **Evidence (Sec. 4).** We provide comprehensive *parameter-level validation* of these *off-principal dynamics*, demonstrating that RLVR exhibits minimal spectral drift, reduced principal-subspace rotation, and distinct avoidance of principal weights compared to SFT.
- **Insight (Sec. 5).** We identify limitations of SFT-era principal-weight masks and principal-component adapter initialization in our tested RLVR settings, *motivating geometry-aware, RLVR-native learning algorithms*.

Table 1: **Update sparsity in SFT vs. RLVR.** Higher $\text{sparsity}_{\text{bf16}}$ indicates more weights unchanged. RLVR is consistently much sparser than SFT, even when SFT is also trained in bf16.

Base Model	Finetuned (FT) Model	Algorithm	Data	Sparsity
Qwen-1.5B	DS-Qwen-1.5B	SFT	Mixed	2.8%
DS-Qwen-1.5B	DeepScaleR-1.5B	GRPO	Math	53.8%
DS-Qwen-1.5B	Archer-Code-1.5B	GRPO	Code	52.5%
DS-Qwen-1.5B	NV-ProRL	GRPO	Mixed†	38.4%
DS-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%
Qwen3-14B-Base	UniReason-SFT	SFT	Math	18.8%
Qwen3-14B-Base	UniReason-RL	DAPO	Math	68.3%
Qwen3-30B-A3B	UloRL-A3B	GRPO	Math	91.7%

Together, these findings provide the first parameter-space account linking RL optimization to weight evolution, complementing concurrent policy-level studies [52, 44], marking a step toward a white-box understanding of RLVR and guiding the development of *geometry-aware* learning methods.

2 A Persistent Optimization Bias in RLVR

We begin with the observation that RL induces sparse updates, going beyond quantification to ask *where RL localizes these changes* to better understand the underlying mechanism. Our analysis reveals that this sparsity stems from a **model-conditioned optimization bias**: for a fixed pretrained model, RL systematically routes visible updates to specific network regions, a phenomenon largely invariant across datasets, RL algorithms, and individual runs. Limited precision (`bfloat16`) attenuates micro-updates in non-preferred regions, rendering the observed sparsity a *superficial readout* of this bias.

Model suite. We evaluate publicly released checkpoints (Tab. 1) spanning multiple RLVR variants (e.g., GRPO, DAPO, Reinforcement++), data domains, and model architectures (Dense vs. MoE). We place particular emphasis on `DeepSeek-R1-Distill-Qwen-1.5B` (`DS-Qwen-1.5B`), for which a long-horizon RL checkpoint is available [23], trained for over 3,000 steps on a *mixed* diverse data mixture encompassing mathematics, coding, STEM, and instruction-following tasks.

2.1 A Robust, `bf16`-aware Update Sparsity Analysis

A rigorous weight probe. `bfloat16` (bf16) is standard in modern RL frameworks like verl [45] to improve throughput without compromising performance. However, its 7 fraction bits mean the spacing between representable values scales with their magnitude. Consequently, fixed absolute-tolerance checks, such as that in [35], can miss small representable changes in BF16-stored weights and overestimate the unchanged fraction (Appendix D.1). We instead use a numerically rigorous probe to define the update sparsity (the fraction of parameters that remain unchanged):

Definition 2.1 (bf16-aware Update Sparsity)**.** Let $\theta^0, \theta^1 \in \mathbb{R}^n$ be finite model parameters stored in `bf16`. For scalars $w, \hat{w}$, they are *unchanged* (denoted $w \approx_\eta^{\text{bf16}} \hat{w}$) iff $|\hat{w} - w| \le \eta \max(|w|, |\hat{w}|)$ with $\eta = 10^{-3}$. This condition detects numerical equality of the stored values, treating $+0$ and -0 as equal (Prop. D.4). Using this, we define the *bf16-aware sparsity* as:

$$\text{sparsity}_{\text{bf16}}(\theta^0, \theta^1) \;:=\; 1 - \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\left[\theta_i^1 \not\approx_\eta^{\text{bf16}} \theta_i^0\right]. \tag{1}$$

Values near 1 indicate few stored changes, while values near 0 indicate dense apparent updates.

RLVR update sparsity results. Tab. 1 confirms that RLVR updates are an order of magnitude sparser (36%–92%) than SFT (0.6%–18.8%). However, these levels are lower than prior reports [35], underscoring the necessity of our rigorous `bf16`-aware probe.

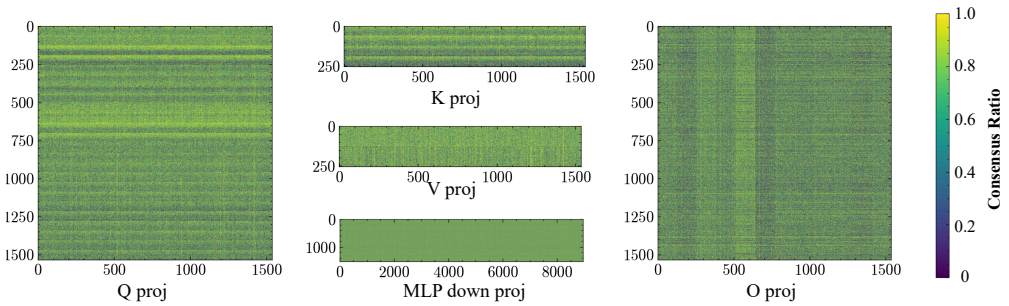

Figure 2: **Consensus ratio of weight updates.** Across five RLVR runs, we plot the 13th layer's projections (Q/K/V/O) and the MLP down projection. Lighter bands mark coordinates updated in most runs, revealing a stable, stripe-like routing pattern rather than random scatter (zoom in to see fine structure).

2.2 RLVR Shows Model-Conditioned Update Locality

Magnitude alone fails to reveal *where* changes occur, obscuring the mechanisms that drive sparsity. To spatially track these updates, we analyze 5 independent RL checkpoints of `DS-Qwen-1.5B` (Tab. 1), covering diverse datasets and algorithms. For each layer ℓ and run r, we compute the `bf16`-aware change mask $M_\ell^{(r)} := \mathbf{1}\big[\, W_\ell^{(r)} \neq_\eta^{\text{bf16}} W_\ell^0 \,\big]$ versus the base weights W_ℓ^0 (Def. D.1).

Stability across runs (Is the bias persistent?) We first quantify spatial agreement via 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 shown in Tab. 2 and Fig. 9, overlap is consistently high across diverse runs, confirming that the base model dictates a shared optimization footprint.

Consensus ratio (Where do updates land?) To determine where these updates concentrate, we 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). As shown in Fig. 2, the consensus maps reveal contiguous bands—localized routing rather than scattered noise. Specifically, we observe prominent row-wise stripes in Q/K/V projections and column-wise stripes in O projections. This structural alignment across runs with disjoint datasets and algorithms starkly confirms the **optimization bias** induced by the base model.

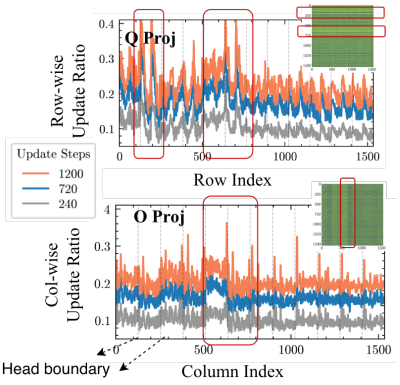

Figure 3: **Temporal emergence of bias.** Row and column-wise update ratios (13th attention block) at steps $t \in \{240, 720, 1200\}$. Consistent with Fig. 2, Q-projections exhibit row-dominance while O-projections are column-dominant. Grey dashed lines show head boundaries, highlighting that the bias emerges both across and *within* heads.

Temporal stability (How does the bias emerge?) To examine *within-run* dynamics, we track the row-wise update 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 step t. In a single `DS-Qwen-1.5B` run, while overall update density increases, the *relative spatial profiles* remain remarkably consistent (Fig. 3). The peaks and troughs align perfectly across steps, indicating that the routing bias *emerges early* and is *reinforced* throughout training, rather than being a transient artifact. In contrast, the orthogonal dimensions (column-wise for Q, row-wise for O) show no discernible trend (Fig. 12), confirming that the bias is indeed structured, not random.

Across model families:(*Is the bias generic?*) We observe similar stripe-structured footprints on `Mistral` and `Llama` (Fig. 11), including a detailed consensus analysis for `Llama` in Appendix F.2. This confirms that the routing bias is generic to RLVR, not unique to `Qwen`.

2.3 Sparsity Is a Superficial Artifact of the Bias

Precision is a filter, not the cause. A natural hypothesis is that `bf16` precision simply truncates small updates. However, this cannot explain the highly structured, model-conditioned routing we observe across disjoint datasets. Furthermore, frameworks like VERL maintain optimizer states and gradient accumulation in `fp32` without truncation. Finite precision filters update *visibility*: a stored parameter changes when its pre-rounding value crosses a rounding-cell boundary (Appendix D.3). The optimization bias governs where updates concentrate; casting can hide changes that remain inside the same cell.

Empirical validation. We validate this via a precision sweep (`bf16`→`fp16`→`fp32`) in Appendix F.3. Visible sparsity decreases sharply with higher precision, while the qualitative geometric signatures of the bias remain invariant: singular-value drift remains small, and principal-subspace rotation remains limited. These results support the precision-lens interpretation of apparent sparsity (Table 4).

Aha Findings! RLVR exhibits a model-conditioned* optimization bias*: updates consistently target specific regions, largely invariant to the dataset or algorithm.

3 A Mechanistic Bridge of RL's Unique Optimization Dynamics

In the post-training era, RL has become a key stage [53]. 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 dynamics with the *Three-Gate Theory*: RL improvement is *anchored* by Gate I (KL Anchor; Sec. 3.1); KL-budgeted motion is *shaped* by pretrained geometry in Gate II (Model Geometry; Sec. 3.2) and then *filtered* by finite precision in Gate III (Precision; Sec. 3.3).

Notations. We consider a large language model with parameters θ, 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: A Local KL Anchor for RL Improvement

Gate I formalizes the *anchoring* effect behind RLVR. At a high level, online RLVR improves the current policy rather than imitating a distant external target as in SFT: reward improvement is realized as a conservative, KL-proximal move. This aligns with recent policy-level evidence that RLVR remains KL-proximal [52, 44]. Our focus is how this constrains the weight changes.

RLVR objective. Many RLVR algorithms, including PPO, GRPO, DAPO, and REINFORCE++, optimize variants of a reward objective with optional reference-policy regularization:

$$\max_\theta \; \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)} R(x,y) - \beta D_{\mathrm{KL}}(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x))\right], \tag{2}$$

where $R(x,y)$ is verifiable and bounded after normalization, π_{ref} is a fixed reference policy, and $\beta \geq 0$ controls explicit reference-KL regularization.

Projected improvement target. Fix a prompt x, let $q(\cdot \mid x)$ be the current or rollout policy, and let $R(x,y)$ be the effective bounded reward signal. For an anchoring strength $\tau > 0$, the ideal KL-regularized improvement target is $\tilde{q}_\tau(y \mid x) \propto q(y \mid x)\exp(R(x,y)/\tau)$. If $\tilde{q}_\tau$ is not exactly representable by the model class, let $\theta^\star$ denote its nearby local M-projection into the policy class.

Proposition 3.1 (Local projected-target KL anchor)**.** *Let*

$$\kappa := D_{\mathrm{KL}}(\tilde{q}_\tau(\cdot \mid x) \,\|\, \pi_\theta(\cdot \mid x)). \tag{3}$$

Under the local regularity and quadratic-growth assumptions stated in Appendix E.2, for κ sufficiently small,

$$D_{\mathrm{KL}}(\pi_{\theta^\star}(\cdot \mid x) \,\|\, \pi_\theta(\cdot \mid x)) \leq \kappa + C\kappa^{3/2}. \tag{4}$$

Thus, when the ideal reward-improvement target is close to the current policy, its projected policy-space target is also close. For a realized RLVR update $\theta^+ = \theta + \Delta$, we use the measured prompt-averaged policy drift

$$D_{\mathcal{D}}(\theta + \Delta, \theta) := \mathbb{E}_{x\sim\mathcal{D}} D_{\mathrm{KL}}(\pi_{\theta+\Delta}(\cdot \mid x) \,\|\, \pi_\theta(\cdot \mid x)) \le K \tag{5}$$

as the empirical small-KL condition connecting practical RLVR runs to the local analysis. This condition is observed in stable and healthy RLVR runs; notably, it also holds in the DAPO without an explicit KL penalty, where the measured KL remains controlled throughout training (Fig. 20).

Proposition 3.2 (Local policy-KL budget ⇒ parameter-norm control)**.** *Let F_θ denote the prompt-averaged Fisher information at θ. Under the local regularity and neighborhood conditions specified in Appendix E.2, if a realized update satisfies Eq. (5) and $F_\theta \succeq \mu I$ on the update subspace for some $\mu > 0$, then as $K \to 0$,*

$$\|\Delta\|_{F_\theta} \le \sqrt{2K}\,(1 + o(1)), \qquad \|\Delta\|_2 \le \sqrt{\frac{2K}{\mu}}\,(1 + o(1)). \tag{6}$$

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

See Appendix E.2 for detailed proofs.

Take-away 1: Gate I converts conservative RL improvement into a local policy-KL budget. Under the stated locality and curvature assumptions, this budget limits parameter motion.

3.2 Gate II: Model Geometry Constrains KL-Budgeted Motion

From Gate I to *location*. While Gate I bounds the update magnitude, it does not specify *where* it lands. We propose **Gate II (Model Geometry)**: unlike random networks, pretrained models possess a structured optimization geometry. Crucially, the local KL geometry makes a given displacement more costly along *high-curvature directions*, motivating a preference for **low-curvature regions**. We examine this proposed **off-principal bias** empirically.

Layerwise Norm Bound. For a linear block W, a restricted curvature lower bound $\mu_W > 0$ and layer-conditioned quadratic budget δ_W give the exact bound $\|\Delta W\|_2 \le \|\Delta W\|_F \le \sqrt{2\delta_W/\mu_W}$ (Lemma E.7 in Appendix E.3).

Conditional spectral stability. Matrix perturbation theory [46] bounds one-step spectral changes under the stated conditions, with $W_0 = W_t$ and $W_+ = W_{t+1}$ in the local KL-based application. Base-to-final stability is assessed empirically in Sec. 4.1.

Limited subspace rotation. First, Theorem 3.3 bounds the rotation of the leading left/right singular subspaces when the update is small relative to the spectral gap. We denote their principal-angle matrices by Θ_U and Θ_V.

Theorem 3.3 (Constrained subspace rotation [50])**.** *Under the layerwise norm bound, let $b_W := \sqrt{2\delta_W/\mu_W}$ and $\gamma_k := \sigma_k(W_0) - \sigma_{k+1}(W_0)$. If $b_W < \gamma_k$, then*

$$\max\left\{\|\sin\Theta_U\|_2, \|\sin\Theta_V\|_2\right\} \le \min\left\{1, \frac{b_W}{\gamma_k - b_W}\right\}. \tag{7}$$

Singular value stability. Second, singular-value changes are bounded by the update norm.

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

$$|\sigma_k(W_+) - \sigma_k(W_0)| \;\le\; \|\Delta W\|_2 \;\le\; \sqrt{\frac{2\delta_W}{\mu_W}};\; \sum_i \big(\sigma_i(W_+) - \sigma_i(W_0)\big)^2 \;\le\; \|\Delta W\|_F^2 \;\le\; \frac{2\delta_W}{\mu_W}. \tag{8}$$

See the detailed proofs in Appendix E.3.

Take-away 2: Under a KL limit, RL updates naturally seek low-curvature directions to avoid abrupt behavioral shifts and preserve the spectral structure (off-principal bias).

Measuring Curvature. Directly quantifying curvature via Hessian in LRMs is challenging, as behavioral shifts are not tied to specific tasks, and capturing full-trajectory KL divergence for long

CoTs is computationally prohibitive. However, the observed sparsity itself implies that the KL-constrained objective naturally suppresses updates in specific directions. To *cheaply* characterize these avoided regions, we adopt a powerful proxy: **principal weights** [26], a subset known to trigger dramatic performance shifts. We validate this proxy via a KL-focused perturbation study (Appendix F.4.2): perturbing the principal subset produces substantially larger KL spikes than perturbing random or off-principal weights consistently across early, middle, and late layers. Direct Fisher probes provide further empirical support for this proxy (Appendix F.4.1).

3.3 Gate III: Precision Acts as a Lens

Building on the optimization bias, limited precision (e.g., `bf16`) acts as a *lens* revealing the bias: it effectively filters out micro-updates in regions where the RL signal is weak, leaving only the geometrically significant moves visible.

Corollary 3.5 (Storage visibility under rounding)**.** *Under round-to-nearest, updates starting from a stored BF16 value have an approximately half-ULP visibility threshold away from binade boundaries; Appendix D.3 gives the exact rounding-cell conditions.*

This quantization mechanism explains the storage-level sparsity discussed in Sec. 2.3. Appendix F.3 reports the corresponding `bf16`/`fp16`/`fp32` comparisons.

4 Theory-Guided Validation of RLVR's Optimization Dynamics

We now empirically validate the Three-Gate Theory. By analyzing spectral drift, subspace rotation, and update location, we demonstrate that RLVR exhibits distinct, off-principal optimization dynamics that set it strikingly apart from SFT.

4.1 RLVR Preserves Spectral Geometry, While SFT Distorts It

We first test whether RL updates respect pretrained geometry by probing spectral changes.

Setups. We analyze checkpoints from an SFT→RLVR pipeline on Qwen3-8B-Base and a long-horizon RL run on `DS-Qwen-1.5B`. We additionally study a setting where SFT and RL are applied separately to Qwen3-14B-Base, matched for in-domain math performance.

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 $\cos\theta_i(U) := \sigma_i\Big(U_{0,k}^\top U_{+,k}\Big)$ and $\cos\theta_i(V) := \sigma_i\Big(V_{0,k}^\top V_{+,k}\Big)$.
- **Spectrum drift.** We show the singular value curve and quantify singular-value change with a normalized ℓ_2 shift: $\mathrm{NSS}(W) = \|\sigma(W_+) - \sigma(W_0)\|_2 / \|\sigma(W_0)\|_2$

Findings. RLVR exhibits *remarkably stable* spectral geometry: across layers, it maintains minimal subspace rotation and singular value profiles nearly identical to initialization. Conversely, SFT induces substantially larger rotations and pronounced spectral drift (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*.

Principal weights as a coordinate proxy. Following [26], $M_{\text{princ}}^{(k)}$ selects the largest-magnitude entries of the rank-k approximation to W_0. These coordinate masks are distinct from spectral directions; their relationship to KL sensitivity is examined empirically (Appendix F.4.2). We also track **low-magnitude weights**, $M_{\text{low}} = \mathrm{Bottom}_\alpha\big(|W_0|\big)$, connecting weight magnitudes to update visibility in Gate III.

Metric. We quantify the alignment between the RL update mask M and a target mask $M_\bullet$ via the overlap ratio: $\mathrm{Overlap}(M_\bullet, M) = |M_\bullet \cap M|/|M|$, compared against the *random guess baseline* as the density of $M_\bullet$ itself, i.e., α.

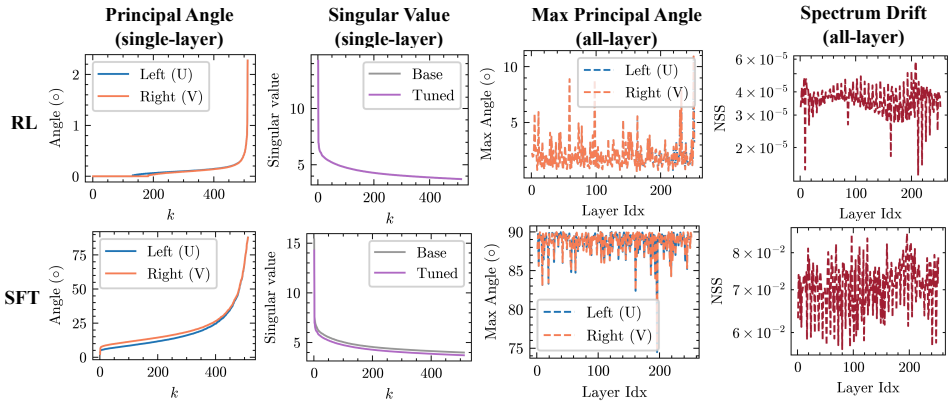

Figure 4: **Spectral geometry under SFT vs. RLVR on Qwen3-8B** [47]. Left: for an exemplar layer, top-k principal angles and singular-value curves. Right: across all layers, maximum principal angle and normalized spectral drift. RLVR maintains a stable top-k spectrum with minimal subspace rotation, unlike SFT. See `DS-Qwen-1.5B` in Fig. 13 and Qwen3-14B-Base in Fig. 14.

Findings. Fig. 5 reveals a clear dichotomy. RL updates exhibit **sub-random overlap** with principal weights, indicating a strong active avoidance. Conversely, updates show **super-random overlap** with low-magnitude weights. Notably, the minor residual overlap with principal weights is driven almost entirely by the intersection of principal and low-magnitude sets; excluding this intersection ($M_{princ} \cap M_{low}^c$) drops the overlap significantly.

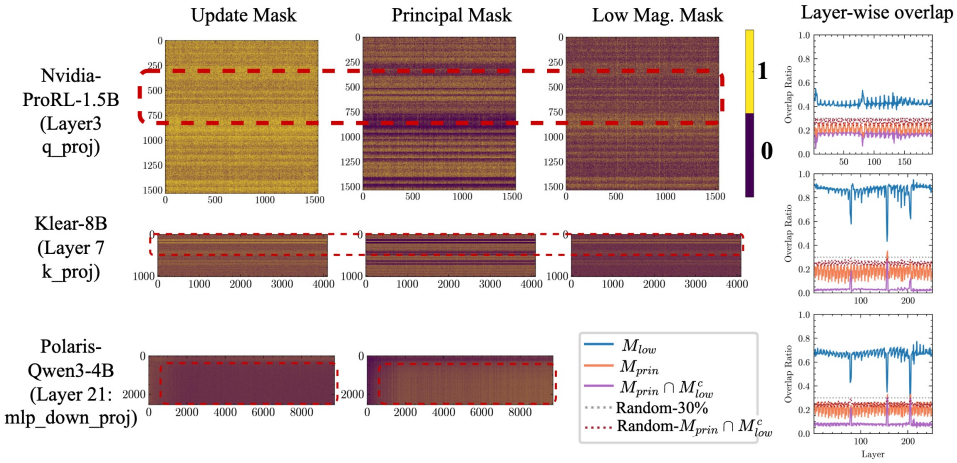

Figure 5: **RL avoids updating principal weights.** We compare the RL update mask with principal weight mask M_{princ}, low magnitude mask M_{low}, and the one $M_{princ} \cap M_{low}^c$. The layer-wise overlap between RL updates and principal weights is consistently *sub-random*, an effect more pronounced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

Insight. This confirms that *RLVR and SFT operate in distinct parameter regimes*. While SFT explicitly leverages the principal backbone as a prior to drive effective optimization [33, 26], RLVR actively avoids high-curvature, principal regions. This fundamental mismatch explains the limited transferability of SFT-centric PEFT heuristics to the RL setting (Sec. 5).

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

Gate II posits that pretrained geometry steers RL updates. To test causality, we *scramble* specific layers of Qwen3-4B-Base via orthogonal rotations on O/V (ROTATE) and head permutations on Q/K/V/O (PERMUTE) (Appendix C), and measure the update overlap $\mathrm{Overlap}(M_\bullet, M) = \frac{|M_\bullet \cap M|}{|M|}$ between a base run vs. another independent run and the intervened run.

Findings. We intervene on (i) layer 20 with ROTATE+PERMUTE and (ii) layer 25 with ROTATE. In Fig. 6, overlap drops to a random level in the intervened layers while remaining high elsewhere, providing causal evidence that pretrained geometry drives the optimization bias.

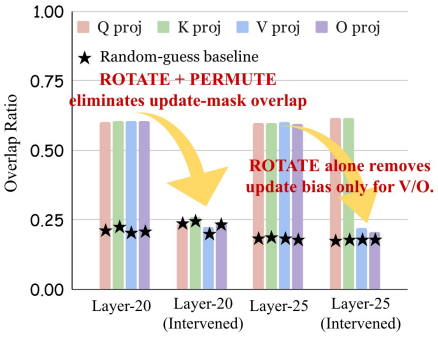

Figure 6: **Overlap ratio** after intervention.

4.4 RLVR Signatures Persist Across Agents, Robotics and Large LR

We extend our diagnostics to agentic, RLHF, and embodied checkpoints in Appendix F.6. We also rule out a simple small-learning-rate artifact in Appendix F.7: even under an aggressive SGD stress test ($\mathrm{lr} = 10^{-2}$), RLVR retains near-zero singular-value drift and sub-degree principal-subspace rotation.

5 Theory-Guided Rethinking of Learning Algorithms for RL

A theory should *explain* and *guide design*. Our analysis reveals that RLVR operates in an off-principal optimization regime, distinct from SFT. This insight directly predicts the failure of many SFT-era PEFT methods that rely on strong principal priors. We validate this by probing sparse and low-rank adaptations, motivating the redesign of geometry-aware learning algorithms.

5.1 Probing Sparse Fine-Tuning in RL

We probe RL's update geometry via **sparse RL fine-tuning**: which weights can be frozen without changing the training trajectory. We build **one-shot masks** from the pretrained model (no extra training) and run sparse training. Following [44], we track the forward KL divergence $\mathrm{KL}(\pi \,\|\, \pi_{\mathrm{ref}})$ between the policy and the base model throughout training. A lag in KL drift relative to the dense baseline indicates that key weights have been pruned, impeding optimization.

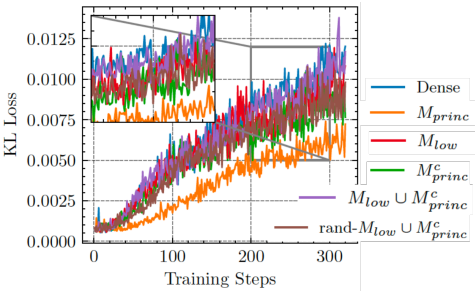

Figure 7: **Forward-KL** under different pretrained masks.

Masks. We construct five masks: (i) M_{princ} (top-50% principal weights, following SFT method [26]), (ii) M^c_{princ} (complementary non-principal coordinate mask), (iii) M_{low} (low-magnitude), (iv) $M_{\mathrm{low}} \cup M^c_{\mathrm{princ}}$ (*our safe mask*, favoring non-principal and low-magnitude weights), and (v) a random baseline matching the sparsity of (iv). We set (i) to 50% sparsity to isolate the geometric effect from parameter count when comparing against (ii).

Findings. $M_{\mathrm{low}} \cup M^c_{\mathrm{princ}}$ best matches the dense RLVR KL curve (Fig. 7). In the 500-step experiment, its average score is 58.38, compared with 58.59 for dense training and 56.96 for the density-matched random mask (Tab. 9). In contrast, the *principal-only* mask M_{princ} yields the slowest KL growth and the weakest performance, confirming that SFT-style principal targeting is misaligned with 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 behavior (KL trajectory and final accuracy) while updating 74.02% of the counted linear-layer 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 [54] is a promising next step.

5.2 Revisiting LoRA Through the Lens of Our Theory

A recent report [42] finds that low-rank LoRA, even rank-1, can match full-parameter RL performance. We investigate whether principal-component initialization, as used by **PiSSA** [33], provides additional benefits under RLVR.

Test. On `DS-Qwen-1.5B` with `DeepMath-103K` [10], we sweep ranks $\{8, 32, 64\}$ and learning rates $\{10^{-4}, 5\times10^{-5}, 10^{-5}\}$ for 200 steps and report pass@1 (mean@16) on AIME24/AMC23 (Fig. 19). We repeat on `Llama-3.2-3B-Instruct` and report pass@1 (mean@4) on MATH500 (Fig. 8).

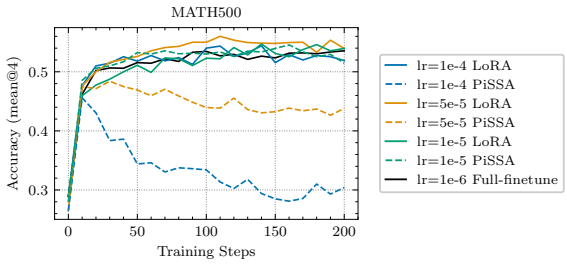

Figure 8: **LoRA vs. PiSSA on `LLaMA-3.2-3B`.** We sweep learning rates $\{1\times10^{-4}, 5\times10^{-5}, 1\times10^{-5}\}$ with a fixed rank of 64 for 200 steps, reporting pass@1 (mean@4) on MATH500. PiSSA initializes its trainable low-rank factors from principal components. Consistent with the `DS-Qwen-1.5B` results in Fig. 19, this initialization provides *no consistent advantage* over LoRA and can become unstable and collapse at higher learning rates in our tested settings.

Findings. PiSSA initializes its trainable low-rank factors from the principal components. In our tested RLVR settings, this initialization provides no consistent advantage over LoRA and can be unstable at high learning rates.

Insight. These results highlight the importance of initialization for RLVR: an initialization motivated by SFT does not necessarily provide the same benefits under RLVR. The observed spectrum-preserving behavior, nearly stable singular values with subspace rotation, suggests that learning update *directions*, as in DoRA [24], may better preserve pretrained structure and align with RLVR.

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 optimization bias* that determines where updates land. We formalized this mechanism with the **Three-Gate Theory**. Empirically, RLVR preserves spectral structure and avoids principal weights, whereas SFT targets them and distorts the spectrum. Beyond explanation, our case studies bridge mechanism and practice: SFT-era principal-weight masks and principal-component adapter initialization show limitations in our tested RLVR settings. 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** learning algorithms. *Limitation.* We diagnose some, rather than fully redesign, the RLVR training stack inspired by the unique optimization dynamics we observe, motivating a broader revisit of commonly used choices inherited from pretraining and SFT.

Acknowledgments and Disclosure of Funding

D. Z. Pan was supported in part by NSF under grant CCF-2112665, the UT Austin iMAGiNE consortium, and an equipment donation from NVIDIA. Z. Wang is supported in part by NSF Awards 2145346 (CAREER) and the NSF AI Institute for Foundations of Machine Learning (IFML). It was also supported by computing support on the Vista GPU Cluster through the Center for Generative AI (CGAI) and the Texas Advanced Computing Center (TACC) at the University of Texas at Austin.

References

[1] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] Cao, S., Hegde, S., Li, D., Griggs, T., Liu, S., Tang, E., Pan, J., Wang, X., Malik, A., Hakhamaneshi, K., Liaw, R., Moritz, P., Zaharia, M., Gonzalez, J. E., and Stoica, I. Skyrl-v0: Train real-world long-horizon agents via reinforcement learning. https://novasky-ai.notion.site/skyrl-v0, 2025. NovaSky AI, Notion page. Accessed 2025-11-10.

[4] Chu, T., Zhai, Y., Yang, J., Tong, S., Xie, S., Schuurmans, D., Le, Q. V., Levine, S., and Ma, Y. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.

[5] Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53, 2024.

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

[7] Dodge, J., Ilharco, G., Schwartz, R., Farhadi, A., Hajishirzi, H., and Smith, N. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*, 2020.

[8] Feng, L., Xue, Z., Liu, T., and An, B. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*, 2025.

[9] Han, S., Pari, J., Gershman, S. J., and Agrawal, P. General reasoning requires learning to reason from the get-go. *arXiv preprint arXiv:2502.19402*, 2025.

[10] He, Z., Liang, T., Xu, J., Liu, Q., Chen, X., Wang, Y., Song, L., Yu, D., Liang, Z., Wang, W., et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

[11] Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[12] Howard, J. and Ruder, S. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[13] Hu, Y., Xie, Q., Jain, V., Francis, J., Patrikar, J., Keetha, N., Kim, S., Xie, Y., Zhang, T., Fang, H.-S., et al. Toward general-purpose robots via foundation models: A survey and meta-analysis. *arXiv preprint arXiv:2312.08782*, 2023.

[14] Huan, M., Li, Y., Zheng, T., Xu, X., Kim, S., Du, M., Poovendran, R., Neubig, G., and Yue, X. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. *arXiv preprint arXiv:2507.00432*, 2025.

[15] Jaech, A. et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[16] Jiang, D., Lu, Y., Li, Z., Lyu, Z., Nie, P., Wang, H., Su, A., Chen, H., Zou, K., Du, C., et al. Verltool: Towards holistic agentic reinforcement learning with tool use. *arXiv preprint arXiv:2509.01055*, 2025.

[17] Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[18] Li, C., Gan, Z., Yang, Z., Yang, J., Li, L., Wang, L., Gao, J., 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.

[19] Li, H., Zuo, Y., Yu, J., Zhang, Y., Yang, Z., Zhang, K., Zhu, X., Zhang, Y., Chen, T., Cui, G., et al. Simplevla-rl: Scaling vla training via reinforcement learning. *arXiv preprint arXiv:2509.09674*, 2025.

[20] Li, Z., Zhang, H., Han, S., Liu, S., Xie, J., Zhang, Y., Choi, Y., Zou, J., and Lu, P. In-the-flow agentic system optimization for effective planning and tool use. *arXiv preprint arXiv:2510.05592*, 2025.

[21] Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

[22] Liu, B., Zhu, Y., Gao, C., Feng, Y., Liu, Q., Zhu, Y., and Stone, P. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36:44776–44791, 2023.

[23] Liu, M., Diao, S., Lu, X., Hu, J., Dong, X., Choi, Y., Kautz, J., and Dong, Y. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025.

[24] Liu, S.-Y., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C. F., Cheng, K.-T., and Chen, M.-H. Dora: Weight-decomposed low-rank adaptation. In *Forty-first International Conference on Machine Learning*, 2024.

[25] Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., and Lin, M. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

[26] Liu, Z., Pang, T., Balabanov, O., Yang, C., Huang, T., Yin, L., Yang, Y., and Liu, S. Lift the veil for the truth: Principal weights emerge after rank reduction for reasoning-focused supervised fine-tuning. *arXiv preprint arXiv:2506.00772*, 2025.

[27] Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[28] Luo, M., Tan, S., Huang, R., Shi, X., Xin, R., Cai, C., Patel, A., Ariyak, A., Wu, Q., Zhang, C., Li, L. E., Popa, R. A., and Stoica, I. 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, 2025. Notion Blog.

[29] Luo, M., Tan, S., Wong, J., Shi, X., Tang, W. Y., Roongta, M., Cai, C., Luo, J., Li, L. E., Popa, R. A., and Stoica, I. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025. Notion Blog.

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

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

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

[33] Meng, F., Wang, Z., and Zhang, M. PiSSA: Principal singular values and singular vectors adaptation of large language models. *Advances in Neural Information Processing Systems*, 37: 121038–121072, 2024.

[34] Meng, Y., Xia, M., and Chen, D. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

[35] Mukherjee, S., Yuan, L., Hakkani-Tur, D., and Peng, H. Reinforcement learning finetunes small subnetworks in large language models. *Advances in Neural Information Processing Systems*, 2025.

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

[37] Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. Improving language understanding by generative pre-training. 2018.

[38] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

[39] Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[40] Raoof, N., Guha, E. K., Marten, R., Mercat, J., Frankel, E., Keh, S., Bansal, H., Smyrnis, G., Nezhurina, M., Vu, T., Sprague, Z. R., Merrill, M. A., Chen, L., Choi, C., Khan, Z., Grover, S., Feuer, B., Suvarna, A., Su, S., Zhao, W., Sharma, K., Ji, C. C.-J., Arora, K., Li, J., Gokaslan, A., Pratt, S. M., Muennighoff, N., Saad-Falcon, J., Yang, J., Aali, A., Pimpalgaonkar, S., Albalak, A., Dave, A., Pouransari, H., Durrett, G., Oh, S., Hashimoto, T., Shankar, V., Choi, Y., Bansal, M., Hegde, C., Heckel, R., Jitsev, J., Sathiamoorthy, M., Dimakis, A., and Schmidt, L. Automatic evals for llms, 2025. URL https://github.com/mlfoundations/evalchemy.

[41] Ross, S., Gordon, G., and Bagnell, D. 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.

[42] Schulman, J. and Lab, T. M. Lora without regret. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250929. https://thinkingmachines.ai/blog/lora/.

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

[44] Shenfeld, I., Pari, J., and Agrawal, P. Rl's razor: Why online reinforcement learning forgets less. *arXiv preprint arXiv:2509.04259*, 2025.

[45] Sheng, G. et al. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*, 2024.

[46] Stewart, G. W. Perturbation theory for the singular value decomposition. 1998.

[47] Su, Z., Pan, L., Bai, X., Liu, D., Dong, G., Huang, J., Hu, W., and Zhou, G. Klear-reasoner: Advancing reasoning capability via gradient-preserving clipping policy optimization. *arXiv preprint arXiv:2508.07629*, 2025.

[48] Team, Q. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[49] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[50] Wedin, P.-Å. Perturbation bounds in connection with singular value decomposition. *BIT Numerical Mathematics*, 12(1):99–111, 1972.

[51] Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021.

[52] Wu, F., Xuan, W., Lu, X., Harchaoui, Z., and Choi, Y. The invisible leash: Why rlvr may not escape its origin. *arXiv preprint arXiv:2507.14843*, 2025.

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

[54] Xiao, Q., Ansell, A., Wu, B., Yin, L., Pechenizkiy, M., Liu, S., and Mocanu, D. C. Leave it to the specialist: Repair sparse llms with sparse fine-tuning via sparsity evolution. *arXiv preprint arXiv:2505.24037*, 2025.

[55] Xiong, W., Yao, J., Xu, Y., Pang, B., Wang, L., Sahoo, D., Li, J., Jiang, N., Zhang, T., Xiong, C., et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025.

[56] Yang, A., Zhang, B., Hui, B., Gao, B., Yu, B., Li, C., Liu, D., Tu, J., Zhou, J., Lin, J., Lu, K., Xue, M., Lin, R., Liu, T., Ren, X., and Zhang, Z. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

[57] Yin, Q., Wu, Y., Shen, Z., Li, S., Wang, Z., Li, Y., Leong, C. T., Kang, J., and Gu, J. Evaluating parameter efficient methods for RLVR. *arXiv preprint arXiv:2512.23165*, 2025.

[58] Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Fan, T., Liu, G., Liu, L., Liu, X., et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[59] Yuan, Y., Cui, H., Huang, Y., Chen, Y., Ni, F., Dong, Z., Li, P., Zheng, Y., and Hao, J. Embodied-r1: Reinforced embodied reasoning for general robotic manipulation. *arXiv preprint arXiv:2508.13998*, 2025.

[60] Zeng, W., Huang, Y., Liu, Q., Liu, W., He, K., Ma, Z., and He, J. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025. URL https://arxiv.org/abs/2503.18892.

[61] Zhai, S., Bai, H., Lin, Z., Pan, J., Tong, P., Zhou, Y., Suhr, A., Xie, S., LeCun, Y., Ma, Y., 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.

[62] Zhang, R., Zhu, J., Zhu, H., and Shi, L. Geometry-preserving orthonormal initialization for low-rank adaptation in RLVR. *arXiv preprint arXiv:2606.31813*, 2026.

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

[64] Zhu, H., Cong, W., Sha, Z., Mukherjee, S., Song, X., González-Martínez, D., Wu, X., Tian, Y., Liu, S., Pan, D. Z., et al. ISO: An RLVR-native optimization stack. *arXiv preprint arXiv:2607.19331*, 2026.

[65] Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Appendix Outline

The appendices are organized as follows.

- **Appendix A: Additional Related Work.** Reviews post-training methods, SFT-versus-RL comparisons, and prior observations of RL parameter updates.
- **Appendix B: Experimental Details.** Describes the models, datasets, training configurations, and benchmark evaluation protocols.
- **Appendix C: Intervention Details.** Details the value/output rotations and attention-head permutations used in the intervention experiments.
- **Appendix D: BF16-Aware Sparsity Analysis.** Explains the limitations of fixed absolute tolerances and the finite-precision basis of the BF16-aware sparsity probe.
- **Appendix E: Mathematical Analysis.** Provides the RLVR objective analysis and detailed proofs for the local KL, parameter-norm, and spectral-perturbation results.
- **Appendix F: Extended Experiments and Visualizations.** Reports additional overlap and precision diagnostics, direct Fisher probes and finite-KL perturbations, spectral diagnostics, cross-domain comparisons, optimizer checks, and LoRA/PiSSA results.

The NeurIPS Paper Checklist follows the technical appendices.

A Additional Related Work

Post-training Large-scale models pre-trained on broad domains serve as general-purpose backbones with extensive domain knowledge and notable zero-shot capabilities [38, 1, 49, 13, 18, 37, 2]. 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 [12, 7, 51, 5], reinforcement learning from human or automated feedback [65, 36, 6, 61], and other recent techniques [39].

Especially, the recent advances in LLM reasoning [6] 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 [36] and GRPO [43]. Building on these successes, a growing body of work [58, 25, 28, 63, 23, 55] 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 [41]. Recent empirical studies consistently reinforce this, finding that RL-tuned models often generalize better out-of-distribution [9, 4] and transfer more effectively to new tasks [14] 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 [35], 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.

Geometry-aware adaptation and optimization for RLVR. Related work extends the study of RLVR update geometry to low-rank initialization and optimization. Zhang et al. [62] develop geometry-preserving orthonormal initialization for low-rank adaptation, while Yin et al. [57] evaluate PEFT methods and SVD-based initializations under RLVR. ISO [64] develops geometry-aware optimization in singular-frame coordinates while preserving pretrained singular values.

B Experimental Details

B.1 Training Settings

Models & Datasets. We run post-training experiments on three open models: **DeepSeek-R1-Distill-Qwen-1.5B** [6], **Qwen2.5-Math-7B** [56], and **Qwen3-Base** [48]. The maximum context length is set to 8192 for DeepSeek-R1-Distill-Qwen-1.5B and Qwen2.5-Math-7B, and to 20480 for 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[1] and the MATH dataset [11]. (2) **DS+SR:** the 47k DeepScaler collection [29] combined with high-difficulty (levels 3–5) problems extracted from SimpleRL [60].We use the version from Huan et al. [14].

Training details. We implement RLVR on the VeRL pipeline [45] (v4.0) and use vLLM [17](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 [58] *without* an explicit reference-KL penalty (ratio clipping as in DAPO), a global batch 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.

[1] DAPO-Math-17k

- **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 [27] (constant learning rate 1×10^{-6}, β_1=0.9, β_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 [47]. We enable an over-length penalty with an additional 1024-token budget and a penalty factor of 1.0.

B.2 Evaluation settings

Since we primarily train on mathematics datasets, we evaluate models on four widely used mathematics benchmarks: AIME24 [31], AIME25 [32], AMC23 [30], and MATH-500 [21]. We use EvalChemy [40] with temperature 0.7 and top-p 0.8. We report **averaged accuracy**, i.e., $pass@1(avg@k)$, to evaluate model performance on all benchmarks. Specifically, we average accuracy over 64 samples per problem for AIME24 and AIME25, 32 for AMC23, and 2 for MATH-500.

C 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

$$R_{kv} = \operatorname{diag}(\underbrace{R,\dots,R}_{H_{kv}}) \in \mathbb{R}^{(H_{kv}D)\times(H_{kv}D)},$$

$$R_q = \operatorname{diag}(\underbrace{R,\dots,R}_{n_{\text{rep}}},\underbrace{R,\dots,R}_{n_{\text{rep}}},\dots) \in \mathbb{R}^{(H_qD)\times(H_qD)}.$$

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

$$\boxed{W_v' = W_v R_{kv}, \qquad W_o' = W_o R_q.} \tag{9}$$

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

Proposition C.1 (Exact invariance)**.** *Let* $\text{Ctx} = \text{Attn}(Q,K,V) \in \mathbb{R}^{\cdot\times(H_qD)}$. *Under equation 9,*

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

Intervention 2: head shuffle (lossless). Order the query heads by their associated KV head, with n_{rep} query heads per group. Let π permute the H_{kv} KV heads, and define $\Pi_{ij} = \mathbf{1}\{i = \pi(j)\}$, so right multiplication places old head $\pi(j)$ at new position j. The corresponding full head-axis permutations are

$$P_{kv} = \Pi \otimes I_D, \qquad P_q = \Pi \otimes I_{n_{\text{rep}}} \otimes I_D.$$

Using the head-in-columns convention $Q = XW_q$, $K = XW_k$, $V = XW_v$, set

$$W_k' = W_kP_{kv}, \qquad W_v' = W_vP_{kv},$$
$$W_q' = W_qP_q, \qquad W_o' = W_oP_q.$$

Any Q/K/V biases follow the same head permutation. Permuting each complete query group together with its KV head gives $\text{Ctx}' = \text{Ctx}P_q$, hence

$$\begin{aligned}\text{out}' &= (\text{Ctx}P_q)(W_oP_q)^\top \\ &= \text{Ctx}P_qP_q^\top W_o^\top = \text{Ctx}W_o^\top.\end{aligned}$$

Equivalently, $(W_o')^\top = P_q^{-1} W_o^\top$: the inverse multiplies the effective output map on the left, while the stored W_o uses $W_o P_q$. For Q/K/V weights stored with output features in rows, the same head indices act on rows; for the stored output projection, they act on columns.

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

D BF16-Aware Sparsity Analysis

D.1 Limits of a Fixed Absolute Tolerance Rule

- **Missed changes in stored BF16 values.** Both $w = 2^{-20}$ and $\widehat{w} = 2^{-20} + 2^{-27}$ are exactly representable in BF16, but their difference is $2^{-27} \approx 7.45 \times 10^{-9} < 10^{-5}$. An absolute-tolerance test at 10^{-5} therefore marks these distinct stored values as unchanged and overestimates the unchanged fraction. For identical finite stored values, their difference is zero, so a nonnegative absolute tolerance cannot create a false change.
- **Pre-quantization values are a different comparison domain.** The higher-precision candidates 1024.001 and 1024.002 differ by $10^{-3} > 10^{-5}$, but both round to 1024 in BF16, whose spacing in this binade is 8. Comparing these candidates before quantization is different from comparing the saved BF16 values; this example does not establish a false positive on BF16-stored inputs.

D.2 Justification of Our Probe

Definition D.1 (Unchanged Weight in bf16)**.** Let $w_i, \widehat{w}_i$ be finite BF16-stored numerical values, including zero. We say w_i is *unchanged* with respect to $\widehat{w}_i$ iff

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

Proposition D.4 establishes numerical equality for this choice of η. Signed zeros are treated as equal; NaNs and infinities are outside this definition.

Lemma D.2 (Gap between distinct bf16 representables)**.** *If $x \neq y$ are positive normal BF16 numbers in the same binade $[2^e, 2^{e+1})$, then*

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

More generally, any numerically distinct finite BF16 values satisfy

$$\frac{|x-y|}{\max(|x|,|y|)} \ge 2^{-8}.$$

Equality is attained across a binade boundary, for example at $x = 1.9921875$ and $y = 2$.

Proof. Within one normal binade, distinct values differ by at least 2^{e-7} and the larger value is below 2^{e+1}, giving the strict bound. For a positive normal value not at a binade boundary, the nearest lower value has the same spacing. At $y = 2^e$, the predecessor gap is at least 2^{e-8}, giving a relative gap of at least 2^{-8}. Adjacent positive subnormal values have relative gap at least $1/127$; nonadjacent pairs only increase the gap for a fixed larger value. Symmetry handles two negative values, while opposite-sign pairs or pairs with one zero have relative gap at least 1. □

Lemma D.3 (ULP lens: magnitude-dependent spacing)**.** *For finite normal BF16 values x with $|x| \in [2^e, 2^{e+1})$, define the binade spacing $\mathrm{ULP}_{\mathrm{bf16}}(x) := 2^{e-7}$. Then*

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

With this spacing convention, $\frac{1}{2}\mathrm{ULP}_{\mathrm{bf16}}(x)/|x| \in [1/510, 1/256]$. This is a spacing statistic; the actual rounding-cell displacement thresholds are given in Appendix D.3.

Proof. Write $|x| = m2^{e-7}$ with $m \in \{128, \ldots, 255\}$. The relative spacing is $1/m$, and its half has minimum $1/(2 \cdot 255) = 1/510$. □

Proposition D.4 (Soundness and completeness of the probe)**.** *Let $w_i, \widehat{w}_i$ be finite BF16-stored values, and suppose $0 \le \eta < 2^{-8}$. Then*

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

Equality is numerical, with $+0$ and -0 identified.

Proof. ($\Rightarrow$) If $w_i \neq \widehat{w}_i$ numerically, their maximum magnitude is positive and Lemma D.2 gives $|\widehat{w}_i - w_i| / \max(|w_i|, |\widehat{w}_i|) \ge 2^{-8} > \eta$, contradicting the test.

($\Leftarrow$) Numerically equal stored values have zero difference and satisfy Eq. (10), including either sign of zero. □

Corollary D.5 (Choice $\eta = 10^{-3}$ is safe)**.** *Since $10^{-3} < 2^{-8}$, Proposition D.4 applies: the test in Eq. (10) passes iff the two finite BF16-stored values are numerically equal. Direct numerical equality of these stored values is an equivalent test; the definition does not equate the two signed-zero bit patterns or recast higher-precision checkpoints into BF16.*

D.3 Rounding Cells and Stored-Update Visibility

Proposition D.6 (Round-to-nearest storage visibility)**.** *Let $\mathcal{Q}_{\mathrm{bf16}}$ denote round-to-nearest to BF16 with a specified tie rule, treating signed zeros as numerically equal. For a finite stored value w with finite adjacent numerical values $w_- < w < w_+$, define*

$$d_- := w - w_-, \qquad d_+ := w_+ - w, \qquad \mathcal{C}_w := \{a : \mathcal{Q}_{\mathrm{bf16}}(a) = w\}.$$

The rounding cell has boundaries $w - d_-/2$ and $w + d_+/2$, with endpoint inclusion determined by the tie rule. If $\mathcal{Q}_{\mathrm{bf16}}(z) = w$, then

$$\mathcal{Q}_{\mathrm{bf16}}(z + \delta) \neq w \quad \Longleftrightarrow \quad z + \delta \notin \mathcal{C}_w.$$

Proof. The midpoints between w and its adjacent representable values are exactly where their distances agree. Round-to-nearest selects w in the interior between these boundaries; the specified tie rule assigns the endpoints. The equivalence follows from the definition of $\mathcal{C}_w$. □

For an update starting from the stored value ($z = w$), the positive and negative distances to the cell boundaries are $d_+/2$ and $d_-/2$. Away from binade boundaries, both gaps equal the local binade spacing, giving the familiar half-ULP threshold; at binade boundaries the gaps can differ. Under ties-to-even, a midpoint is assigned to the adjacent value with an even least-significant significand bit.

If a higher-precision master value z is retained, the distances to the right and left cell boundaries are $w + d_+/2 - z$ and $z - w + d_-/2$. Thus visibility depends on the master offset $z - w$, as well as on the update δ. FP32 optimizer moments alone do not establish that an FP32 master parameter is retained. The endpoint-equality probe compares stored values; its separation bound is distinct from these continuous rounding-cell displacement thresholds.

E Mathematical Analysis

E.1 Practical RLVR Surrogates

This subsection records the sampled objectives used by practical RLVR implementations. It is included for implementation context only; it is not used to prove the Gate-I KL anchor in Prop. 3.1. The main theory instead analyzes the ideal KL-regularized improvement oracle and connects realized optimizer steps through measured policy KL.

Let π_{old} be the rollout policy used to sample G completions $y^{1:G}$ for a prompt x. For token t in completion i, define the importance ratio

$$r_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\mathrm{old}}(y_{i,t} \mid x, y_{i,<t})}.$$

A generic PPO/GRPO/DAPO-style clipped surrogate can be written as

$$\mathcal{J}_{\text{clip}}(\theta) = \mathbb{E}_{x,\, y^{1:G}\sim\pi_{\text{old}}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\min\Big(r_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{i,t}\Big)\right],$$

where $\hat{A}_{i,t}$ is an estimated, normalized, or group-normalized advantage. Different RLVR variants differ in how $\hat{A}_{i,t}$ is computed, whether reference-KL shaping is added, and whether clipping is applied at the token, sequence, or objective level.

In the main theory, these implementation details are abstracted into an effective bounded reward signal R and a local reference policy q, typically the current or rollout policy. A realized optimizer step $\theta^+ = \theta + \Delta$ is not assumed to solve the ideal projection problem; it enters the theory through its measured policy KL.

E.2 Proof of Gate I: Local KL Anchor for RL Improvement

This appendix proves Prop. 3.1 and Prop. 3.2. The proof has two parts. First, a KL-regularized oracle in distribution space is projected into the policy class, giving a local projected-target KL anchor. Second, under the stated locality and restricted-curvature assumptions, a realized update satisfying a measured policy-KL budget has controlled Fisher and Euclidean norm. This appendix does not claim that a practical PPO/GRPO/DAPO optimizer step exactly solves the projection problem.

Throughout the Gate-I projection proof, fix a prompt x. Let $q(\cdot \mid x)$ have full support on $\mathcal{Y}$, let $R(x, y)$ be bounded, and let $\tau > 0$. We suppress conditioning on x when it is clear.

1. Distribution-space oracle and policy-space projection.

Lemma E.1 (KL-regularized oracle)**.** *The optimization problem*

$$\max_{p\in\Delta(\mathcal{Y})}\{\mathbb{E}_{y\sim p}[R(y)] - \tau D_{\text{KL}}(p\|q)\}$$

has the unique maximizer

$$\tilde{q}_\tau(y) = \frac{q(y)\exp(R(y)/\tau)}{Z_\tau}, \qquad Z_\tau = \sum_{y\in\mathcal{Y}} q(y)\exp(R(y)/\tau).$$

Moreover,

$$\mathbb{E}_p[R] - \tau D_{\text{KL}}(p\|q) = \tau\log Z_\tau - \tau D_{\text{KL}}(p\|\tilde{q}_\tau).$$

Proof. For any $p \in \Delta(\mathcal{Y})$,

$$\begin{aligned}
\mathbb{E}_p[R] - \tau D_{\text{KL}}(p\|q) &= \sum_y p(y)R(y) - \tau\sum_y p(y)\log\frac{p(y)}{q(y)} \\
&= -\tau\sum_y p(y)\log\frac{p(y)}{q(y)\exp(R(y)/\tau)} \\
&= \tau\log Z_\tau - \tau\sum_y p(y)\log\frac{p(y)}{\tilde{q}_\tau(y)} \\
&= \tau\log Z_\tau - \tau D_{\text{KL}}(p\|\tilde{q}_\tau).
\end{aligned}$$

Since KL divergence is nonnegative and vanishes only when $p = \tilde{q}_\tau$, the maximizer is unique. □

Lemma E.2 (M-projection as policy-space representation)**.** *For fixed $\tilde{q}_\tau$,*

$$\arg\min_\vartheta D_{\text{KL}}(\tilde{q}_\tau\|\pi_\vartheta) = \arg\max_\vartheta \mathbb{E}_{y\sim\tilde{q}_\tau}[\log\pi_\vartheta(y \mid x)].$$

Proof. The KL objective decomposes as

$$D_{\text{KL}}(\tilde{q}_\tau\|\pi_\vartheta) = \mathbb{E}_{\tilde{q}_\tau}[\log\tilde{q}_\tau] - \mathbb{E}_{\tilde{q}_\tau}[\log\pi_\vartheta],$$

where the first term is independent of ϑ. □

The M-projection identifies the ideal nearby policy-space target,

$$\theta^\star \in \arg\min_{\vartheta} D_{\mathrm{KL}}(\tilde{q}_\tau \| \pi_\vartheta).$$

It is a representation step for the oracle target, not a claim that a practical mini-batch optimizer step solves this problem.

2. Local projected-target KL anchor. Let

$$f(\vartheta) := D_{\mathrm{KL}}(\tilde{q}_\tau \| \pi_\vartheta), \qquad \kappa := f(\theta).$$

For each y, write

$$s_\vartheta(y) := \nabla_\vartheta \log \pi_\vartheta(y \mid x), \qquad H_\vartheta(y) := \nabla^2_\vartheta \log \pi_\vartheta(y \mid x).$$

Assumptions. We use the following local assumptions for Prop. 3.1. There is an open convex neighborhood U containing θ such that:

(A1) For every y, $\vartheta \mapsto \log \pi_\vartheta(y \mid x)$ is C^3 on U, with uniformly bounded second and third derivatives.

(A2) There exists a nearby local M-projection $\theta^\star \in U$ minimizing f over U.

(A3) f has local quadratic growth at $\theta^\star$: there exists $m > 0$ such that, for all $\vartheta \in U$,

$$f(\vartheta) \geq f(\theta^\star) + \frac{m}{2}\|\vartheta - \theta^\star\|_2^2.$$

Let

$$h := \theta - \theta^\star.$$

Define

$$H_{\tilde{q}}(\theta^\star) := -\mathbb{E}_{y\sim\tilde{q}_\tau}[H_{\theta^\star}(y)], \qquad F_{\theta^\star} := \mathbb{E}_{y\sim\pi_{\theta^\star}}[s_{\theta^\star}(y)s_{\theta^\star}(y)^\top] = -\mathbb{E}_{y\sim\pi_{\theta^\star}}[H_{\theta^\star}(y)].$$

Lemma E.3 (Local expansions at the projected target)**.** *Under (A1)–(A2), there exist constants $C_f, C_g < \infty$ such that*

$$f(\theta) - f(\theta^\star) = \frac{1}{2}h^\top H_{\tilde{q}}(\theta^\star)h + R_f, \qquad |R_f| \leq C_f\|h\|_2^3,$$

and

$$D_{\mathrm{KL}}(\pi_{\theta^\star}\|\pi_\theta) = \frac{1}{2}h^\top F_{\theta^\star}h + R_g, \qquad |R_g| \leq C_g\|h\|_2^3.$$

Proof. Since $\theta^\star$ minimizes f over the open set U, it is an interior critical point: $\nabla f(\theta^\star) = 0$. Taylor expansion of f at $\theta^\star$ gives the first expansion with a cubic remainder.

For the second expansion, define

$$g(\vartheta) := D_{\mathrm{KL}}(\pi_{\theta^\star}\|\pi_\vartheta).$$

Then $g(\theta^\star) = 0$ and

$$\nabla g(\theta^\star) = -\mathbb{E}_{\pi_{\theta^\star}}[s_{\theta^\star}(Y)] = 0.$$

Moreover,

$$\nabla^2 g(\theta^\star) = -\mathbb{E}_{\pi_{\theta^\star}}[H_{\theta^\star}(Y)] = F_{\theta^\star}.$$

A Taylor expansion of g at $\theta^\star$ gives the second expansion. □

Lemma E.4 (Hessian–Fisher mismatch)**.** *Under (A1), for a constant $L_2 < \infty$ bounding $\|H_\vartheta(y)\|_{\mathrm{op}}$ on U,*

$$\|H_{\tilde{q}}(\theta^\star) - F_{\theta^\star}\|_{\mathrm{op}} \leq L_2\sqrt{2\kappa}.$$

Consequently,

$$\left|h^\top\left(H_{\tilde{q}}(\theta^\star) - F_{\theta^\star}\right)h\right| \leq L_2\sqrt{2\kappa}\,\|h\|_2^2.$$

Proof. By definition,

$$H_{\tilde{q}}(\theta^\star) - F_{\theta^\star} = -\sum_y \left(\tilde{q}_\tau(y) - \pi_{\theta^\star}(y \mid x)\right) H_{\theta^\star}(y).$$

Thus

$$\|H_{\tilde{q}}(\theta^\star) - F_{\theta^\star}\|_{\mathrm{op}} \le L_2 \|\tilde{q}_\tau - \pi_{\theta^\star}\|_1.$$

Since $\theta^\star$ minimizes f over U and $\theta \in U$,

$$D_{\mathrm{KL}}(\tilde{q}_\tau \| \pi_{\theta^\star}) \le D_{\mathrm{KL}}(\tilde{q}_\tau \| \pi_\theta) = \kappa.$$

Pinsker's inequality gives

$$\|\tilde{q}_\tau - \pi_{\theta^\star}\|_1 \le \sqrt{2\kappa}.$$

The quadratic-form bound follows immediately. □

Proof of Prop. 3.1. By local quadratic growth (A3),

$$\frac{m}{2}\|h\|_2^2 \le f(\theta) - f(\theta^\star) \le f(\theta) = \kappa.$$

Therefore

$$\|h\|_2^2 \le \frac{2\kappa}{m}, \qquad \|h\|_2^3 \le \left(\frac{2}{m}\right)^{3/2} \kappa^{3/2}.$$

Subtracting the two expansions in Lemma E.3 yields

$$D_{\mathrm{KL}}(\pi_{\theta^\star} \| \pi_\theta) - (f(\theta) - f(\theta^\star)) = \frac{1}{2} h^\top \left(F_{\theta^\star} - H_{\tilde{q}}(\theta^\star)\right) h + (R_g - R_f).$$

By Lemma E.4, the quadratic mismatch term is $O(\kappa^{3/2})$. The remainders are also $O(\kappa^{3/2})$. Hence, for some constant $C < \infty$,

$$D_{\mathrm{KL}}(\pi_{\theta^\star} \| \pi_\theta) \le f(\theta) - f(\theta^\star) + C\kappa^{3/2} \le \kappa + C\kappa^{3/2},$$

because $f(\theta^\star) \ge 0$. □

3. Local policy-KL budget and parameter-norm control. Now let $\theta^+ = \theta + \Delta$ be a realized optimizer update. Define the prompt-averaged policy drift

$$D_{\mathcal{D}}(\theta + \Delta, \theta) := \mathbb{E}_{x \sim \mathcal{D}} D_{\mathrm{KL}}\left(\pi_{\theta+\Delta}(\cdot \mid x) \,\|\, \pi_\theta(\cdot \mid x)\right).$$

Let the prompt-averaged Fisher be

$$F_\theta := \mathbb{E}_{x \sim \mathcal{D},\, y \sim \pi_\theta(\cdot|x)}\left[\nabla_\theta \log \pi_\theta(y \mid x) \nabla_\theta \log \pi_\theta(y \mid x)^\top\right].$$

Lemma E.5 (Prompt-averaged quadratic KL model)**.** *Assume common positive support and local C^3 regularity sufficient to differentiate the per-prompt KL and average its derivatives. For some radius $r_0 > 0$, assume the third derivatives of the per-prompt KL are uniformly bounded over prompts whenever $\|\Delta\|_2 \le r_0$. These regularity conditions can be imposed on a finite empirical prompt set with a finite sequence horizon, or justified by suitable dominated-derivative bounds. Then there is a constant $C \ge 0$ such that, throughout this neighborhood,*

$$D_{\mathcal{D}}(\theta + \Delta, \theta) = \frac{1}{2}\Delta^\top F_\theta \Delta + R(\Delta), \qquad |R(\Delta)| \le C\|\Delta\|_2^3.$$

Proof. For each fixed prompt x, normalization gives zero KL and zero first derivative at $\Delta = 0$, and the KL Hessian there is $F_\theta(x)$. Taylor's theorem with the uniform third-derivative bound therefore gives

$$D_{\mathrm{KL}}\left(\pi_{\theta+\Delta}(\cdot \mid x) \,\|\, \pi_\theta(\cdot \mid x)\right) = \frac{1}{2}\Delta^\top F_\theta(x)\Delta + R_x(\Delta), \qquad |R_x(\Delta)| \le C\|\Delta\|_2^3,$$

where

$$F_\theta(x) = \mathbb{E}_{y \sim \pi_\theta(\cdot|x)}\left[\nabla_\theta \log \pi_\theta(y \mid x) \nabla_\theta \log \pi_\theta(y \mid x)^\top\right].$$

The averaging assumptions justify $F_\theta = \mathbb{E}_{x \sim \mathcal{D}} F_\theta(x)$. Averaging the uniformly controlled remainders over $x \sim \mathcal{D}$ gives the result. □

Locality and restricted curvature. For Prop. 3.2, consider an update subspace $\mathcal{S}$ on which $\Delta^\top F_\theta \Delta \geq \mu \|\Delta\|_2^2$ for some $\mu > 0$ and all $\Delta \in \mathcal{S}$. With r_0, C from Lemma E.5, assume the realized update belongs to $\mathcal{S}$ and satisfies

$$\|\Delta\|_2 \leq r_{\mathrm{loc}}, \qquad r_{\mathrm{loc}} := \begin{cases} \min\{r_0, \mu/(4C)\}, & C > 0, \\ r_0, & C = 0. \end{cases}$$

Keep r_0, C, μ fixed, or uniformly controlled with μ bounded away from zero, as $K \to 0$. These are sufficient conditions on the analyzed local update family; the positive curvature bound is imposed only on the stated subspace. The locality condition is an explicit assumption and does not follow from small policy KL alone.

Proof of Prop. 3.2. Let

$$a := \Delta^\top F_\theta \Delta.$$

By Lemma E.5,

$$D_{\mathcal{D}}(\theta + \Delta, \theta) = \frac{1}{2} a + R(\Delta), \qquad |R(\Delta)| \leq C \|\Delta\|_2^3$$

for the constant C in the locality condition. The restricted Fisher lower bound and the assumed neighborhood radius imply

$$D_{\mathcal{D}}(\theta + \Delta, \theta) \geq \left(\frac{\mu}{2} - C \|\Delta\|_2 \right) \|\Delta\|_2^2 \geq \frac{\mu}{4} \|\Delta\|_2^2.$$

Combining this with $D_{\mathcal{D}}(\theta + \Delta, \theta) \leq K$ yields $\|\Delta\|_2^2 \leq 4K/\mu$. Substituting this bound back into the Taylor expansion gives

$$\begin{aligned} a &= 2 D_{\mathcal{D}}(\theta + \Delta, \theta) - 2R(\Delta) \\ &\leq 2K + 2C \|\Delta\|_2^3 \\ &\leq 2K + \frac{16C}{\mu^{3/2}} K^{3/2} = 2K\big(1 + O(\sqrt{K})\big). \end{aligned}$$

The uniform control of the constants makes the relative remainder $o(1)$ as $K \to 0$. Thus

$$\|\Delta\|_{F_\theta} = \sqrt{\Delta^\top F_\theta \Delta} \leq \sqrt{2K}\,(1 + o(1)).$$

Using $F_\theta \succeq \mu I$ again,

$$\|\Delta\|_2 \leq \sqrt{\frac{\Delta^\top F_\theta \Delta}{\mu}} \leq \sqrt{\frac{2K}{\mu}}\,(1 + o(1)).$$

For any weight block $W \subset \theta$, $\|\Delta W\|_F$ is the Euclidean norm of a subvector of Δ, so

$$\|\Delta W\|_F \leq \|\Delta\|_2 \leq \sqrt{\frac{2K}{\mu}}\,(1 + o(1)).$$

□

4. What clipping alone does and does not prove. A true pointwise density-ratio trust region would imply a KL bound.

Proposition E.6 (Pointwise trust region implies a KL bound)**.** *Let p and q be strictly positive distributions on a finite set. Suppose that, for some $\epsilon \in (0, 1)$,*

$$1 - \epsilon \leq \frac{p(y)}{q(y)} \leq 1 + \epsilon \qquad \textit{for all } y.$$

Then

$$D_{\mathrm{KL}}(p \| q) \leq \frac{\epsilon^2}{2(1 - \epsilon)}.$$

Proof. Let $r(y) := p(y)/q(y)$ and define

$$\phi(r) := r \log r - r + 1.$$

Since $\mathbb{E}_q[r] = 1$,

$$D_{\mathrm{KL}}(p\|q) = \mathbb{E}_q[r \log r] = \mathbb{E}_q[\phi(r)].$$

On $r \in [1-\epsilon, 1+\epsilon]$, we have

$$\phi(1) = 0, \qquad \phi'(1) = 0, \qquad \phi''(r) = \frac{1}{r} \le \frac{1}{1-\epsilon}.$$

Taylor's theorem gives

$$\phi(r) \le \frac{(r-1)^2}{2(1-\epsilon)} \le \frac{\epsilon^2}{2(1-\epsilon)}.$$

Taking expectation under q proves the claim. □

PPO/DAPO-style clipping is applied to sampled surrogate ratios and does not imply the pointwise density-ratio assumption above over the full output distribution. Therefore clip-only DAPO is not used as a theorem-level global KL trust region in our analysis. In the main text, Fig. 20 is used only as empirical evidence that a stable clip-only run can remain in the measured small-KL regime required by Gate I.

E.3 Proofs for Gate II (Sec. 3.2)

Setup (layer-conditioned budget). At the reference iterate θ_t, consider a single update $\theta_{t+1} = \theta_t + \Delta\theta$. For its linear block, $W_0 = W_t$ and $W_+ = W_{t+1}$ in the local KL-based application. Write $u := \mathrm{vec}(\Delta W)$, $v := \Delta\theta_{\neg W}$, and partition the prompt-averaged Fisher as

$$F_{\theta_t} = \begin{bmatrix} A & B \\ B^\top & C \end{bmatrix} \succeq 0, \qquad Q(u,v) := \frac{1}{2}\begin{bmatrix} u \\ v \end{bmatrix}^\top F_{\theta_t} \begin{bmatrix} u \\ v \end{bmatrix}.$$

Let $C^\dagger$ denote the Moore–Penrose pseudoinverse. Positive semidefiniteness implies $Bz = 0$ for $z \in \ker C$, hence $\mathrm{range}(B^\top) \subseteq \mathrm{range}(C)$. The generalized Schur complement and layer-conditioned quadratic budget are

$$S_W := A - BC^\dagger B^\top \succeq 0, \qquad \delta_W := \inf_{\widetilde{v}} Q(u, \widetilde{v}) = \frac{1}{2} u^\top S_W u.$$

Indeed, completing the square gives

$$Q(u,v) = \delta_W + \frac{1}{2}\big(v + C^\dagger B^\top u\big)^\top C \big(v + C^\dagger B^\top u\big).$$

Thus S_W accounts for compensation by other parameters, whereas the fixed-other-parameter curvature block is A. Assume a restricted curvature lower bound on a specified update subspace $\mathcal{U}_W$:

$$u \in \mathcal{U}_W, \qquad z^\top S_W z \ge \mu_W \|z\|_2^2 \quad \text{for all } z \in \mathcal{U}_W, \qquad \mu_W > 0.$$

Under the local assumptions of Prop. 3.2 and Lemma E.5, the actual policy KL satisfies

$$D_{\mathcal{D}}(\theta_t + \Delta\theta, \theta_t) = Q(u,v) + R(\Delta\theta), \qquad |R(\Delta\theta)| \le C_3 \|\Delta\theta\|_2^3.$$

Consequently, a measured KL budget $D_{\mathcal{D}}(\theta_t + \Delta\theta, \theta_t) \le K$ gives

$$0 \le \delta_W \le Q(u,v) \le K + C_3 \|\Delta\theta\|_2^3 \le K + O(K^{3/2}),$$

where the last estimate uses the local parameter-norm control in Prop. 3.2 with uniformly controlled constants. Each δ_W is a minimized conditional quadratic cost; the budgets for different layers need not sum to K.

Lemma E.7 (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 $u \in \mathcal{U}_W$, the restricted curvature bound gives $\delta_W \ge \frac{1}{2}\mu_W \|u\|_2^2 = \frac{1}{2}\mu_W \|\Delta W\|_F^2$. The operator norm is bounded by the Frobenius norm. □

Lemma E.8 (Wedin's sin–Θ). *Let $W_0 \in \mathbb{R}^{m\times n}$, $W_+ = W_0 + E$, and $1 \le k < \min(m,n)$. Write $\varepsilon_W := \|E\|_2$ and $\gamma_k := \sigma_k(W_0) - \sigma_{k+1}(W_0)$, and assume $\varepsilon_W < \gamma_k$. For*

$$\Theta_U := \Theta\big(U_k(W_0), U_k(W_+)\big), \qquad \Theta_V := \Theta\big(V_k(W_0), V_k(W_+)\big),$$

the leading left/right singular subspaces satisfy

$$\max\big\{\|\sin\Theta_U\|_2, \|\sin\Theta_V\|_2\big\} \le \min\left\{1, \frac{\varepsilon_W}{\gamma_k - \varepsilon_W}\right\}.$$

Proof. Choose complete orthonormal singular bases $U = [U_1\ U_2]$ and $V = [V_1\ V_2]$ of W_0, with U_1, V_1 containing its leading k singular vectors. The complementary block $A_2 := U_2^\top W_0 V_2$ satisfies $\|A_2\|_2 = \sigma_{k+1}(W_0)$. Let $(\widehat{U}_1, \widehat{\Sigma}_1, \widehat{V}_1)$ be leading k singular triplets of W_+ and set $X := U_2^\top \widehat{U}_1$, $Y := V_2^\top \widehat{V}_1$. The singular-vector equations give

$$X\widehat{\Sigma}_1 = A_2 Y + U_2^\top E \widehat{V}_1,$$
$$Y\widehat{\Sigma}_1 = A_2^\top X + V_2^\top E^\top \widehat{U}_1.$$

Weyl's inequality yields $\sigma_k(W_+) \ge \sigma_k(W_0) - \varepsilon_W > \sigma_{k+1}(W_0)$, so $\widehat{\Sigma}_1$ is invertible. Multiplying the residual equations by $\widehat{\Sigma}_1^{-1}$ and taking operator norms, with $M := \max\{\|X\|_2, \|Y\|_2\}$, gives

$$\big(\sigma_k(W_+) - \sigma_{k+1}(W_0)\big) M \le \varepsilon_W.$$

The mixed-spectrum separation is at least $\gamma_k - \varepsilon_W > 0$, hence $M \le \varepsilon_W/(\gamma_k - \varepsilon_W)$. Finally, $\|X\|_2 = \|\sin\Theta_U\|_2$, $\|Y\|_2 = \|\sin\Theta_V\|_2$, and $M \le 1$. □

Lemma E.9 (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 E.10 (Projection stability). *Assume the layerwise curvature bound in Lemma E.7, with $\mu_W > 0$ on the stated update subspace, and let $b_W := \sqrt{2\delta_W/\mu_W} < \gamma_k$. With Θ_U, Θ_V as in Lemma E.8,*

$$\big\|U_k(W_0)U_k(W_0)^\top - U_k(W_+)U_k(W_+)^\top\big\|_2 = \|\sin\Theta_U\|_2 \le \min\left\{1, \frac{b_W}{\gamma_k - b_W}\right\}.$$

The analogous bound holds for the right subspaces with V_k. In particular, if $b_W \le \gamma_k/2$, the upper bound is at most $2b_W/\gamma_k$.

Proof. Lemma E.7 gives $\varepsilon_W = \|\Delta W\|_2 \le b_W < \gamma_k$. Apply Lemma E.8 and use that $t/(\gamma_k - t)$ is increasing on $[0, \gamma_k)$. The projector–angle equality is exact for equal-dimensional subspaces. This also proves Theorem 3.3. □

Interpretation. Small rotation is guaranteed when b_W/γ_k is sufficiently small; in this regime the bound is $O\big(\sqrt{\delta_W/\mu_W}/\gamma_k\big)$. The local KL-based application takes $W_0 = W_t$ and $W_+ = W_{t+1}$; this one-step guarantee does not by itself bound cumulative drift. The base-to-final comparisons in Sec. 4.1 assess spectral stability empirically.

Corollary E.11 (Top-k energy and Ky Fan norms). *Let $\|\cdot\|_{(k)} := \sum_{i=1}^k \sigma_i(\cdot)$ be the Ky Fan k-norm. Then*

$$\begin{aligned} \big|\,\|W_+\|_{(k)} - \|W_0\|_{(k)}\,\big| &\le \sum_{i=1}^k |\sigma_i(W_+) - \sigma_i(W_0)| \\ &\le k\,\|\Delta W\|_2 \le k\sqrt{\tfrac{2\delta_W}{\mu_W}}. \end{aligned} \tag{11}$$

F Extended Experiments and Visualizations

F.1 Spatial Consistency Analysis (Jaccard Overlap)

To rigorously quantify the spatial consistency of updates across independent training runs, we analyze the pairwise Jaccard similarity between the final update masks. Fig. 9 visualizes the aggregate agreement matrix, while Tab. 2 details the specific overlap statistics.

Despite the runs utilizing disjoint datasets (Math, Code, Mixed) and different algorithms, we observe consistently high similarity scores that far exceed the expected baseline for random sparse updates. This confirms that the update sparsity is not random; rather, the optimization process targets a shared, deterministic subset of parameters dictated by the pre-trained model structure.

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

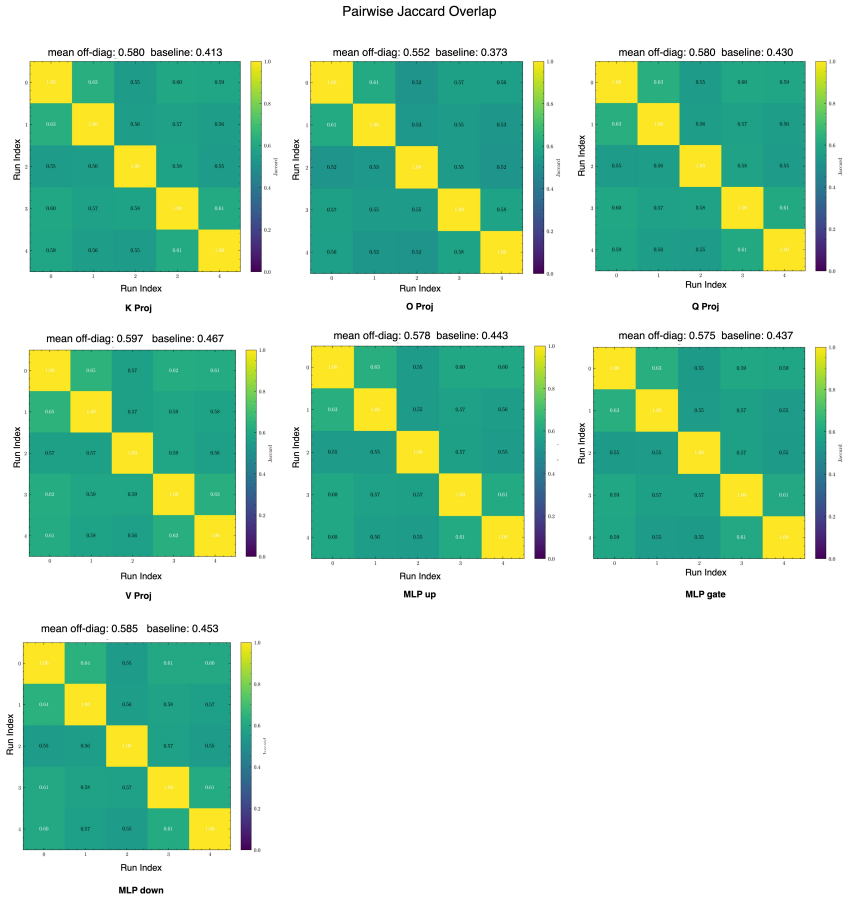

Figure 9: Pair-wise Jaccard similarity of update masks from five independent RLVR runs on Layer 13 of the `DS-Distill-Qwen-1.5B` model.

F.2 Consensus ratio and Algorithm Robustness

To further validate our findings on the "Implicit Compass," we extended our analysis to **Llama-3.2-3B-Instruct** across five distinct reinforcement learning configurations. We varied the algorithm (Majority-Voting, Self-Certainty, Co-rewarding, GRPO) and the dataset (MATH vs. DAPO-14k). The resulting sparsity levels are detailed in Tab. 3.

Table 3: **Update sparsity of Llama-3.2-3B-Instruct across reinforcement learning algorithms and training datasets.** Checkpoint source: the Co-rewarding collection.

Dataset	Method	Sparsity
MATH	Majority-Voting	71.21%
MATH	Self-Certainty	83.24%
MATH	Co-rewarding	71.86%
MATH	GRPO	71.28%
DAPO-14k	GRPO	66.00%

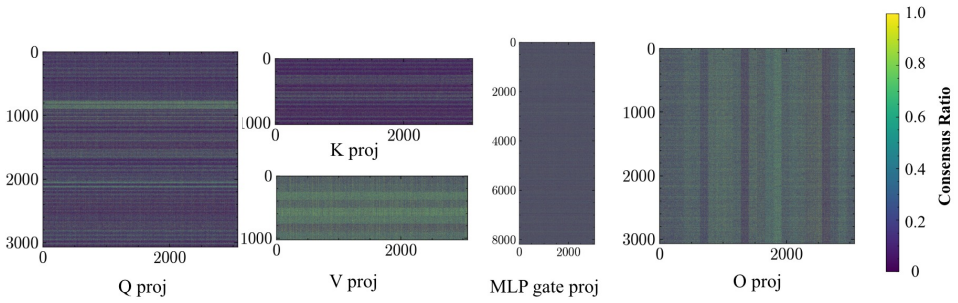

Figure 10: **Robustness of Consensus across Algorithms.** Visualization of the update consensus masks for Llama-3.2-3B-Instruct across five different runs using distinct RL algorithms (Majority-Voting, Self-Certainty, Co-rewarding, GRPO) and datasets (MATH, DAPO-14k). The consistent observation of "stripe-like" structures (row-wise in Q/K/V, column-wise in O) confirms that the "Implicit Compass" is intrinsic to the model's optimization dynamics and robust to the choice of reinforcement learning method.

Same Model, Different Algorithms (Consensus Check): We applied the consensus ratio metric $C_{\ell,ij}$ to these new runs. As shown in Fig. 10, we observe the identical "stripe-like" consensus in update masks. Even without changing the task, the fact that independent RL runs (using disjoint data and variants like Self-Certainty or Co-rewarding) consistently route updates to the same regions confirms the "Implicit Compass" is intrinsic to the optimization dynamics, not a artifact of a specific model.

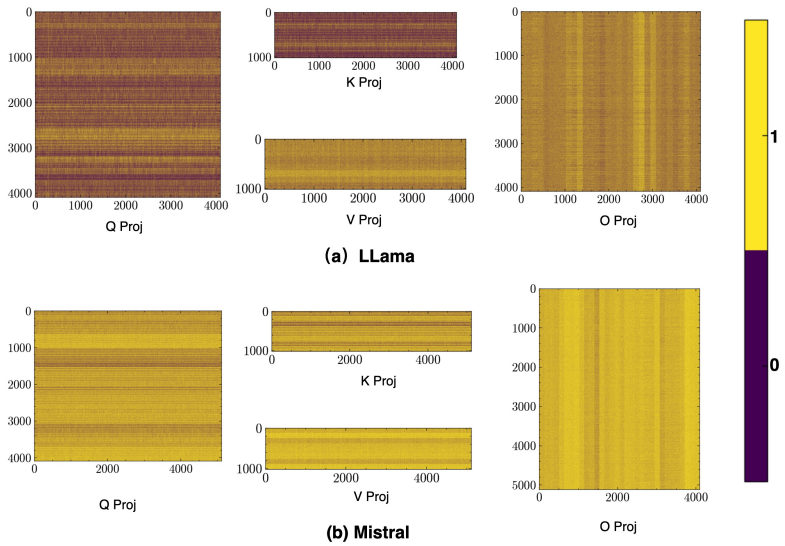

Figure 11: Structured Update observed on Llama(Llama-3.1-8B) and Mistral (Mistral-Small-24B) models. Here we plot the weight update mask using the zero-RL checkpoints from [60].

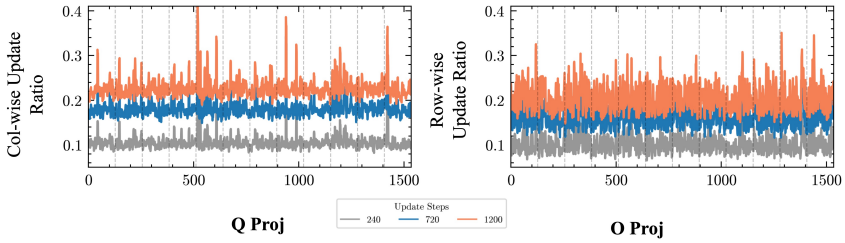

Figure 12: Temporal emergence of the optimization bias with row and column-wise update ratios for the 13th attention block across gradient update steps ($t \in \{240, 720, 1200\}$), smoothed with a 3-step window. *The column-wise (Q) and row-wise (O) update ratios show a much weaker bias.*

F.3 Precision sweep

To determine whether the observed sparsity is a physical property of the optimization or a numerical artifact, we conduct a strict precision sweep on `DS-Qwen-1.5B` (trained on DeepMath-103K [10]) and `Llama-3.2-3B` (trained on MATH), using our standard BF16 setup as a baseline.

Precision Sweep (BF16 → FP16 → FP32). The results in Tab. 4 support the precision-lens interpretation: apparent sparsity decreases with higher precision, while the qualitative geometric signatures remain invariant:

- **Sparsity (Update Visibility):** Whole-model sparsity decreases from $82.38\% \rightarrow 35.16\% \rightarrow 3.25\%$ for DS-Qwen and $68.76\% \rightarrow 18.65\% \rightarrow 0.00\%$ for Llama across BF16, FP16, and FP32. This is consistent with higher precision revealing micro-updates that BF16 leaves invisible in stored weights.
- **Geometry (Invariant Qualitative Signatures):** The qualitative signatures of spectral stability remain invariant across the tested precisions, with small singular-value drift and limited principal-subspace rotation. Reported maximum singular-value drifts range from 6.40×10^{-5} to 3.29×10^{-4}. DS-Qwen's reported angles remain below $1.6°$, while Llama's MLP-down maximum decreases from approximately $15.52° \rightarrow 5.43° \rightarrow 2.01°$ across BF16, FP16, and FP32. The numerical magnitudes vary across models, modules, and precisions, while the overall qualitative pattern of spectral stability persists.

Table 4: **Precision sweep: invariant qualitative geometric signatures.** Increasing precision sharply reduces *apparent* sparsity (fraction of unchanged stored parameters). The qualitative signatures of spectral stability remain invariant across the tested precisions, although their numerical magnitudes vary across models and modules. Here $\Delta\sigma_{\max}$ denotes the maximum (layerwise) singular-value drift for the probed linear blocks, and $\theta_{\max}$ denotes the maximum top-k principal-subspace rotation (see Sec. 4 for metric definitions).

DS-Qwen-1.5B

Module	Metric	bf16	fp16	fp32
Whole model	Sparsity (%)	82.38%	35.16%	3.25%
Q proj	Sparsity (%)	79.78%	26.89%	0.00%
	Max θ Rot. ($^\circ$)	0.4731/0.4735	0.7871/0.7881	0.6215/0.6209
	Max σ Diff	8.90E-05	1.65E-04	1.54E-04
K proj	Sparsity (%)	77.40%	26.17%	0.00%
	Max θ Rot. ($^\circ$)	0.0685/0.3626	0.0626/0.4049	0.0685/0.4182
	Max σ Diff	7.90E-05	1.27E-04	1.44E-04
V proj	Sparsity (%)	76.89%	24.67%	0.00%
	Max θ Rot. ($^\circ$)	0.0593/0.2846	0.0626/0.3263	0.0626/0.4073
	Max σ Diff	8.70E-05	1.16E-04	1.48E-04
O proj	Sparsity (%)	83.60%	32.54%	0.00%
	Max θ Rot. ($^\circ$)	0.3375/0.3386	1.3442/1.3467	0.5279/0.5403
	Max σ Diff	6.40E-05	9.60E-05	1.16E-04
MLP Up	Sparsity (%)	80.63%	27.00%	0.00%
	Max θ Rot. ($^\circ$)	0.8339/0.8330	0.7836/0.7833	1.2045/1.2037
	Max σ Diff	9.70E-05	1.50E-04	2.26E-04
MLP Down	Sparsity (%)	79.85%	26.17%	0.00%
	Max θ Rot. ($^\circ$)	0.6543/0.6555	1.4175/1.4180	1.5846/1.5847
	Max σ Diff	1.37E-04	1.50E-04	2.27E-04
MLP Gate	Sparsity (%)	81.61%	28.05%	0.00%
	Max θ Rot. ($^\circ$)	0.3690/0.3696	0.6299/0.6284	0.5795/0.5774
	Max σ Diff	9.10E-05	1.63E-04	1.68E-04

Table 4 (continued): Llama-3.2-3B.

Module	Metric	bf16	fp16	fp32
Whole model	Sparsity (%)	68.76%	18.65%	0.00%
Q proj	Sparsity (%)	68.20%	16.02%	0.00%
	Max θ Rot. ($^\circ$)	0.7838/0.7826	1.0099/1.0107	0.7426/0.7431
	Max σ Diff	1.99E-04	2.29E-04	3.29E-04
K proj	Sparsity (%)	74.10%	20.40%	0.00%
	Max θ Rot. ($^\circ$)	2.0905/2.0921	2.2828/2.2847	1.9078/1.9082
	Max σ Diff	2.97E-04	2.99E-04	1.66E-04
V proj	Sparsity (%)	51.32%	8.82%	0.00%
	Max θ Rot. ($^\circ$)	1.9608/1.9654	1.3452/1.3430	1.9078/1.9141
	Max σ Diff	1.26E-04	1.39E-04	1.40E-04
O proj	Sparsity (%)	56.17%	10.20%	0.00%
	Max θ Rot. ($^\circ$)	3.2303/3.2325	2.9191/2.9209	1.5612/1.5706
	Max σ Diff	1.00E-04	8.30E-05	9.50E-05
MLP Up	Sparsity (%)	62.98%	12.65%	0.00%
	Max θ Rot. ($^\circ$)	3.4001/3.3988	3.3315/3.3315	1.5000/1.5012
	Max σ Diff	1.23E-04	1.48E-04	2.69E-04
MLP Down	Sparsity (%)	62.18%	12.36%	0.00%
	Max θ Rot. ($^\circ$)	15.5249/15.5248	5.4301/5.4302	2.0066/2.0066
	Max σ Diff	2.13E-04	2.43E-04	3.26E-04
MLP Gate	Sparsity (%)	63.63%	13.01%	0.00%
	Max θ Rot. ($^\circ$)	2.6317/2.6305	1.6107/1.6084	1.3662/1.3640
	Max σ Diff	1.72E-04	2.02E-04	2.13E-04

F.4 Direct Fisher Probes of Gate II

F.4.1 Direct Fisher Measurements

Gate II relates local policy sensitivity to the amount of parameter motion permitted by a KL budget. We examine this link using direct policy Fisher estimates, distinguishing the curvature of pretrained spectral directions from that of the principal coordinate masks used in Fig. 5. These probes complement the finite-KL perturbation analysis below.

Setup and local Fisher. We probe one checkpoint of Qwen3-1.7B-Base using on-policy rollouts and an identical frozen copy of the policy as the local reference. Let ρ denote the probe distribution of token histories, held fixed when differentiating the candidate parameters θ'. Define

$$\mathcal{K}_\rho(\theta, \theta') = \mathbb{E}_{h_t \sim \rho}[D_{\mathrm{KL}}(\pi_\theta(\cdot \mid h_t) \,\|\, \pi_{\theta'}(\cdot \mid h_t))]\,. \tag{12}$$

With the token score $g = \nabla_\theta \log \pi_\theta(y_t \mid h_t)$, where $y_t \sim \pi_\theta(\cdot \mid h_t)$, the corresponding Fisher is $F_\theta = \mathbb{E}_{h_t, y_t}[gg^\top]$. At self-reference,

$$\nabla_{\theta'} \mathcal{K}_\rho(\theta, \theta')\big|_{\theta'=\theta} = 0, \qquad \nabla^2_{\theta'} \mathcal{K}_\rho(\theta, \theta')\big|_{\theta'=\theta} = F_\theta. \tag{13}$$

We estimate the relevant Fisher quantities using 1,536 on-policy gradient samples, covering all 196 linear matrices. Directional and diagonal quantities can be estimated from score samples without materializing the dense Fisher. A symmetric-KL perturbation check supports the local quadratic approximation at the tested probe scales.

Curvature of leading spectral directions. For each probed matrix W, let $W_0 = \sum_i \sigma_i u_i v_i^\top$ be its pretrained counterpart, and define the paired singular directions $B_i = u_i v_i^\top$, with $\|B_i\|_F = 1$. Let F_W be the block of F_θ indexed by the entries of W. Holding other parameters fixed, the curvature of B_i is

$$\begin{aligned} c(B_i) &= \mathrm{vec}(B_i)^\top F_W \,\mathrm{vec}(B_i) \\ &= \mathbb{E}\Big[\langle B_i, \nabla_W \log \pi_\theta(y_t \mid h_t)\rangle_F^2\Big]. \end{aligned} \tag{14}$$

We average curvature over the leading k modes and compare against equal-norm magnitude-matched non-principal and isotropic random controls. Table 5 reports the resulting ratios.

Table 5: Fisher curvature ratios for leading spectral directions. Each column compares the leading-k average against the indicated equal-norm control. Bracketed values are the reported interval estimates.

Leading k	Magnitude-matched non-principal control	Isotropic random control
1	20.1 [15.9, 25.6]	16.1 [11.4, 21.9]
4	10.3 [8.0, 12.8]	7.9 [6.0, 10.9]
8	6.8 [5.2, 9.4]	5.6 [4.2, 7.2]
16	4.6 [3.7, 6.5]	3.8 [2.9, 4.7]
32	3.2 [2.6, 4.1]	2.5 [2.1, 3.0]
64	2.2 [1.9, 2.6]	1.7 [1.5, 1.9]

The leading directions have greater mean curvature than both tested controls, with the strongest enrichment at $k = 1$. These averages summarize the probed paired directions; curvature of their linear combinations also depends on Fisher cross terms. To interpret the directional result geometrically, let $\theta(a; B)$ change only W to $W + aB$, for a unit-Frobenius direction B. Locally,

$$\mathcal{K}_\rho\big(\theta, \theta(a; B)\big) = \tfrac{1}{2} a^2 c(B) + o(a^2). \tag{15}$$

In this quadratic approximation, the amplitude permitted by a fixed KL budget scales as $c(B)^{-1/2}$. For example, a curvature ratio of 20.1 corresponds to an amplitude ratio of $1/\sqrt{20.1} \approx 0.22$ at equal KL cost. This is a pure-direction comparison; it is not a measurement of the relative amplitudes of actual optimizer updates.

Ordinary coordinate Fisher. The spectral directions above are matrix-valued perturbations. The principal mask in Fig. 5 instead selects coordinates according to the magnitudes of entries in a rank-k reconstruction. We therefore evaluate coordinate curvature separately, using

$$\kappa_F(M) = \frac{1}{|M|} \sum_{j \in M} (F_W)_{jj}. \tag{16}$$

Table 6 compares the principal mask with a magnitude-matched non-principal mask. Here q is the fraction of coordinates selected by the mask, the density denoted by α in Sec. 4.2; we evaluate $q = 5\%$ and $q = 30\%$.

Table 6: Principal-to-control ratios of ordinary coordinate Fisher for two coordinate-mask densities q. Ratios below one indicate no positive enrichment for that setting.

Mask density q	$k = 1$	$k = 4$	$k = 8$	$k = 16$
5%	2.54	1.91	1.58	1.32
30%	1.65	1.31	0.99	0.85

Coordinate enrichment is weaker than spectral-direction enrichment and decreases with k over these settings. At $q = 30\%$, the ratios for $k = 8$ and $k = 16$ are 0.99 and 0.85, respectively. Thus the coordinate proxy is supported selectively by ordinary Fisher curvature; the 20.1-fold spectral-direction result does not characterize all principal coordinates.

Curvature at a nominal BF16 visibility scale. Fig. 5 concerns updates visible in stored BF16 weights. To connect coordinate curvature to this scale, we use the visibility-weighted score

$$\tau_j = \tfrac{1}{2}\,\mathrm{ULP}_{\mathrm{BF16}}(W_j), \qquad s_j^{\mathrm{vis}} = \tau_j^2 (F_W)_{jj}, \qquad \kappa_{\mathrm{vis}}(M) = \frac{1}{|M|} \sum_{j \in M} s_j^{\mathrm{vis}}. \tag{17}$$

Here τ_j is a nominal half-ULP scale. As detailed in Appendix D.3, the exact displacement needed to change a stored value depends on the rounding cell, the direction of motion, the tie rule, and any offset of a retained higher-precision master value. We therefore interpret s_j^{vis} as a scale-aware diagnostic. For a single-coordinate perturbation, half this score is the leading quadratic KL term evaluated at the nominal scale. Averaging these scores summarizes individual-coordinate costs; the KL cost of a simultaneous masked update also depends on Fisher cross terms.

Let R_{vis} be the principal-to-control ratio of the visibility-weighted score, and let R_{mag} be the corresponding magnitude-only baseline ratio. We report both quantities and the normalized ratio $R_{\mathrm{excess}} = R_{\mathrm{vis}}/R_{\mathrm{mag}}$ in Table 7. This normalization assesses enrichment relative to the magnitude-only baseline.

Table 7: Visibility-weighted coordinate Fisher at mask density $q = 30\%$. Ratios compare the principal mask with the magnitude-matched non-principal control. The final column reports enrichment after magnitude normalization; brackets retain the reported interval estimates.

k	Magnitude-only R_{mag}	Measured R_{vis}	Excess R_{excess}
1	1.04	1.93	1.85 [1.65, 2.08]
8	1.06	1.67	1.58 [1.42, 1.75]
16	1.07	1.61	1.50 [1.35, 1.67]
32	1.10	1.57	1.42 [1.28, 1.59]
64	1.18	1.55	1.31 [1.19, 1.46]

The measured visibility-weighted ratios range from 1.93 to 1.55, and the magnitude-normalized excess ratios range from 1.85 to 1.31. All tested settings show positive enrichment in this visibility-weighted diagnostic, including settings where ordinary coordinate Fisher enrichment is weak. This provides a resolution-aware interpretation of the principal-coordinate avoidance measured in Fig. 5.

Interpretation and scope. The direct probes support two distinct findings: strong curvature enrichment along the leading pretrained spectral directions, and weaker coordinate enrichment that is more consistently expressed at the nominal BF16 visibility scale. Together they support the local curvature

premise of Gate II and connect it to the visibility diagnostic used for Gate III. This evidence comes from one model scale and one probe checkpoint. Relating these fixed-other-parameter probes to general joint updates requires consistent KL normalization and the conditional-curvature assumptions in Appendix E.3; spectral preservation over training is assessed by the empirical spectral measurements elsewhere in the paper.

F.4.2 Finite-KL Perturbations of Coordinate Masks

We complement the direct Fisher probes above with finite-KL perturbations of selected coordinate masks.

Experimental Setup. We measure the *change in KL divergence* ($\Delta\mathrm{KL}$) induced by perturbing specific subsets of weights. To ensure our findings are robust and not artifacts of specific depths or module types, we employ a **stratified sampling strategy** spanning the model's architecture:

- **Depth Coverage:** We select **Layer 2** (Early/Embedding-aligned), **Layer 13** (Middle/Reasoning-core), and **Layer 23** (Late/Output-aligned).
- **Module Diversity:** Within each layer, we random test on Attention heads (**Q_proj, V_proj**) and MLP blocks (**Gate_proj, Up_proj**), which have distinct spectral properties.

Methodology. For each selected module W, we select principal-weight coordinates from the largest-magnitude entries of its rank-k approximation. We perturb (a) these **principal-weight coordinates**, (b) their **non-principal coordinate complement**, or (c) **randomly selected coordinates**. We then evaluate the model on the AI-MO/NuminaMath-CoT and record the shift in KL divergence relative to the base model. We choose 30% to keep the same number of parameters while adding noise following the same procedure in [26].

Results. As shown in Tab. 8, perturbing the principal weights consistently triggers a much larger spike in KL divergence compared to random perturbations. Crucially, this trend holds **universally** across all sampled layers (2, 13, 23) and module types (Q, V, Gate, Up).

Table 8: **Finite-KL Perturbation Analysis.** Ratio of KL divergence change ($\Delta\mathrm{KL}$) after injecting Gaussian noise into different coordinate masks. Principal-weight masks show higher sensitivity than random or off-principal coordinate controls across the selected early, middle, and late layers.

Layer Index	**Module**	$\Delta\mathrm{KL}$ **Ratio (Principal / Random)**	$\Delta\mathrm{KL}$ **Ratio (Principal / Off-Principal)**
Layer 2	mlp.gate_proj	3.427	6.233
	self_attn.v_proj	1.324	3.793
Layer 13	mlp.up_proj	1.437	1.778
	self_attn.q_proj	2.590	5.098
Layer 23	mlp.up_proj	1.558	2.244
	self_attn.v_proj	1.996	8.392

F.5 Spectrum shift for DS-1.5B and Qwen3-14B

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

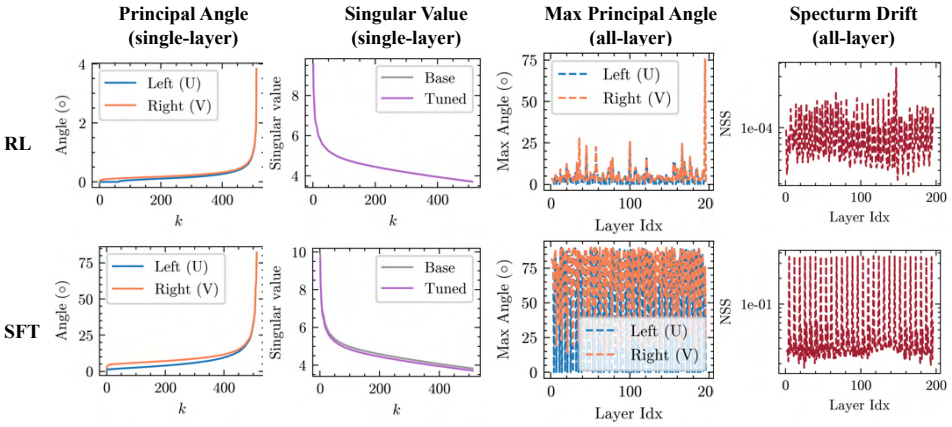

Figure 13: The spectrum probe results on the RL and SFT version on the `DS-Distill-Qwen-1.5B` [23]. RLVR shows surprisingly stable top-k spectrum with minimal subspace rotation and top-k singular value changes.

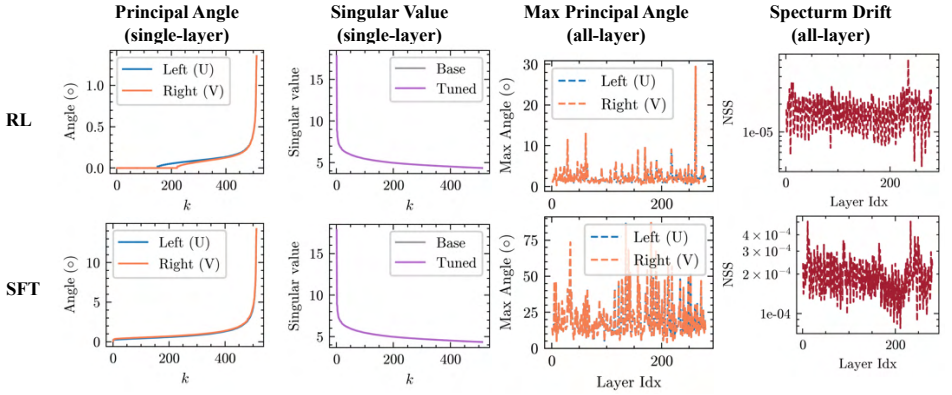

Figure 14: The spectrum probe results on the RL and SFT version on the Qwen3-14B [14]. RLVR shows surprisingly stable top-k spectrum with minimal subspace rotation and top-k singular value changes.

Table 9: Performance of DS-Qwen-1.5B with different masking strategies after 500 training steps. Average is the unweighted mean of the four displayed benchmark scores, rounded to two decimal places. Parameter counts are for linear layers only, excluding the embedding and head layers. Detailed evaluation settings are available in Appendix B.2. ***The safe mask*** $M^c_{princ} \cup M_{low}$ ***has the highest average among the masked runs; principal-only training has the lowest average.***

DS-Qwen-1.5B Mask	**Math500**	**AMC23**	**AIME24**	**AIME25**	**Average**	**#params**
Dense	**84.5**	**83.52**	38.28	**28.075**	**58.59**	100%
M_{princ}	83.60	78.83	34.06	25.63	55.53	50%
M^c_{princ}	84.0	77.97	38.64	27.81	57.11	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	58.38	74.02%
Random-$M^c_{princ} \cup M_{low}$	84.10	81.72	34.69	27.34	56.96	74.02%

F.6 RLVR signatures persist in agentic tasks, RLHF, robotics

Table 10: **Model List for analyzed checkpoints for agentic and embodied AI (robotics manipulation) tasks and RLHF algorithms.**

Base Model	FT Model	Algorithm	Data	Sparsity
Agent				
Qwen3-8B	SkyRL-Agent-WebResearch-8B	GRPO	WebResearch	40.56%
Qwen3-8B	VT-deepsearch-8B	GRPO	Deepsearch	89.67%
Qwen3-8B	VT-SWE-8B	GRPO	SWE	84.32%
Qwen2.5-7B-Instruct	agentflow-planner-7b	Flow-GRPO	Planning	80.99%
Qwen2.5-7B-Instruct	GiGPO-Qwen2.5-7B-Instruct-WebShop	GiGPO	WebShop	51.7%
Qwen2.5-7B-Instruct	GiGPO-Qwen2.5-7B-Instruct-ALFWorld	GiGPO	ALFWorld	62.08%
RLHF				
Meta-Llama-3-8B-Instruct	Llama-3-Instruct-8B-DPO	DPO	instruction-following	82.38%
Meta-Llama-3-8B-Instruct	Llama-3-Instruct-8B-SimPO	SimPO	instruction-following	71.00%
Embodied AI				
openvla/openvla-7b	Openvla-oft-SFT-libero10-trajall	SFT	Robotic manipulation Liu et al. [22]	3.46%
Openvla-oft-SFT-libero10-traj1	openvla-oft-libero10-traj1-rl	GRPO	Robotic manipulation Liu et al. [22]	35.04%
Qwen2.5-VL-3B-Instruct	Embodied-R1-3B-v1	GRPO	Robotic Manipulation	44.28%

Setup. We analyze additional *agent*, *RLHF* (RL with human feedback), and *embodied* checkpoints and apply the same weight–space diagnostics as in Sec. 4.1 and Sec. 4.2: (i) principal-subspace rotation, (ii) spectral drift, and (iii) update–principal misalignment. The extended model suite is summarized in Tab. 10. **(i)Agents.** We evaluate policies from AGENTFLOW [20] and VERL-AGENT [8] on multi-turn and long-horizon tasks. We also assess tool-augmented agents from SKYRL [3] and VERL-TOOL [16] on *WebSearch*, *DeepSearch*, and *SWE*. **(ii)RLHF.** We include preference-optimized models trained with DPO [39] and SimPO [34], primarily targeting instruction following. **(iii)Embodied AI.** We include embodied AI models with both language and vision backbones from SimpleVLA-RL [19] and Embodied R1 [59].

Our Findings. **(i) Stable spectra, mostly small rotations.** Layer spectra remain near-identical to the base model, and most probed blocks show small rotations, with larger outliers in some blocks (Fig. 15; Fig. 16; Fig. 17), consistent with the earlier off-principal pattern. **(ii) Off-principal updates.** Update masks in checkpoints consistently *avoid principal weights*: the most active bands are spatially misaligned with the principal mask (Fig. 18). **Takeaway.** RLVR's optimization dynamics, *mostly small rotations, spectrum preservation, off-principal routing*, persist beyond verifiable math/code to *agents*, *embodied AI RLHF*, indicating a common, model-conditioned optimization bias within a KL-anchored RL post-training game, consistent with our Three-Gate Theory.

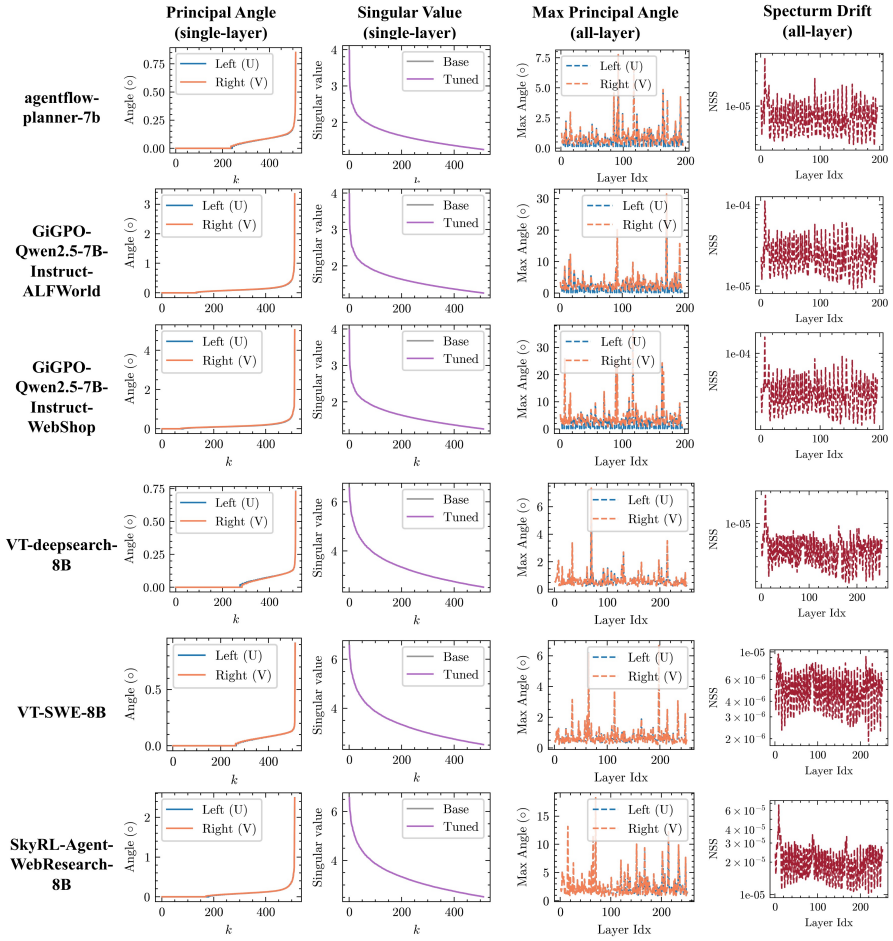

Figure 15: **Spectrum under RL in agent tasks.** In agent settings, including multi-turn interactions and tool use, RL leaves layer singular-value spectra nearly unchanged and induces only small rotations of the top-k singular subspaces, consistent with the spectrum-preserving, off-principal RLVR regime. Results for RL with human feedback (RLHF), which exhibit the same optimization signature, appear in Fig. 16. For consistency, we use the *second block O*-projection layer as an exemplar single-layer readout.

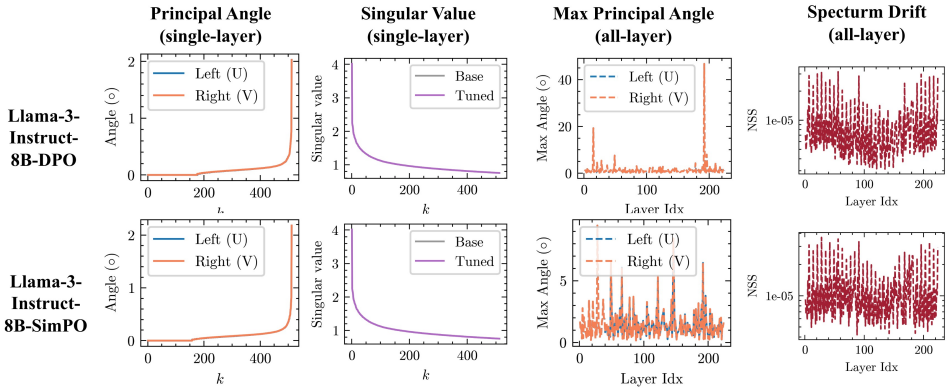

Figure 16: **Spectral geometry under RLHF setting [34].** Across RLHF checkpoints, RL training preserves layer spectra and induces only minor rotation of the top-k subspaces, consistent with the RLVR regime.

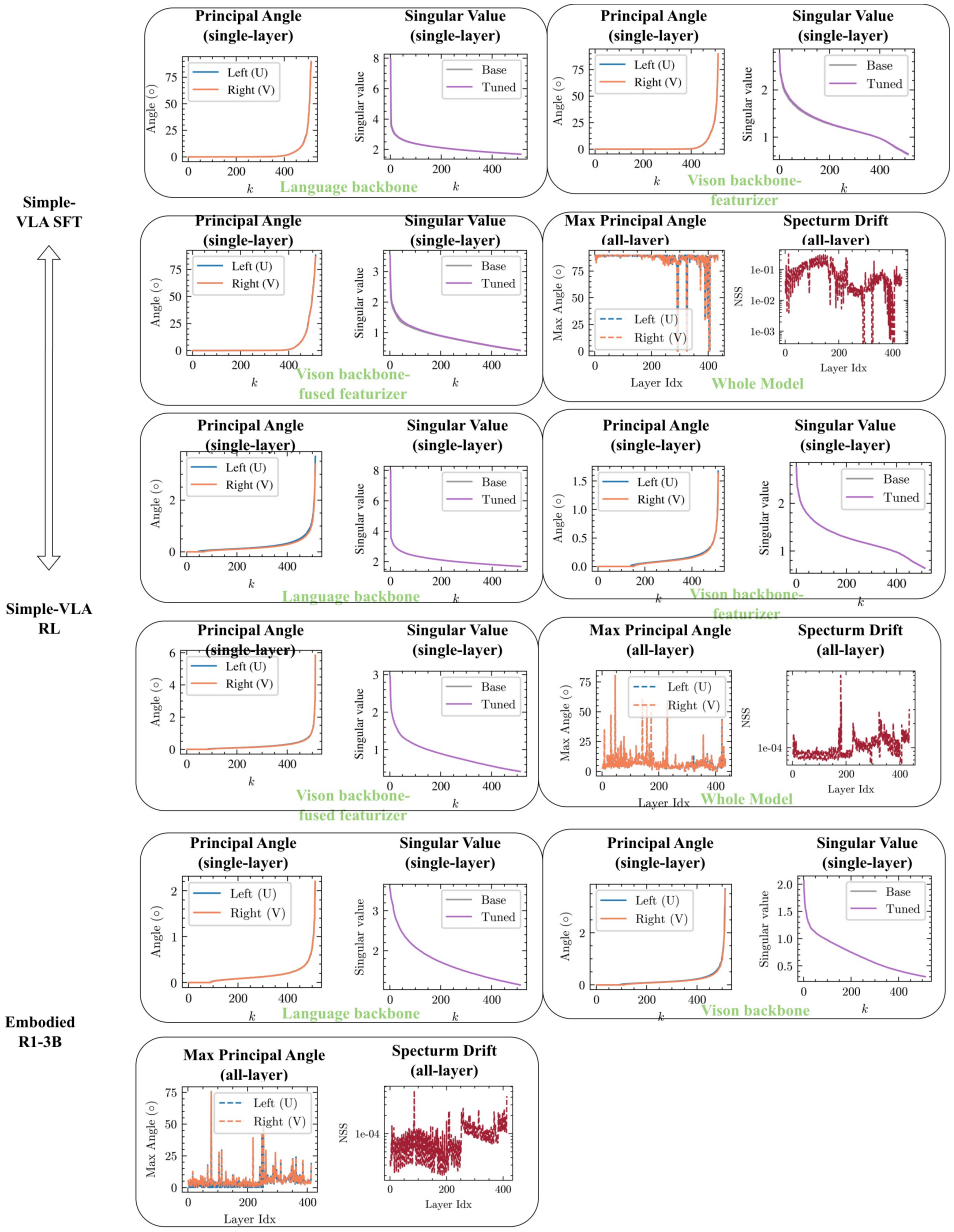

Figure 17: **Spectral geometry under embodied AI tasks: SimpleVLA-RL [19] and Embodied R1 [59].** Across the language and vision backbones, RL training preserves layer spectra; most probed blocks show small rotations, with larger outliers in some blocks.

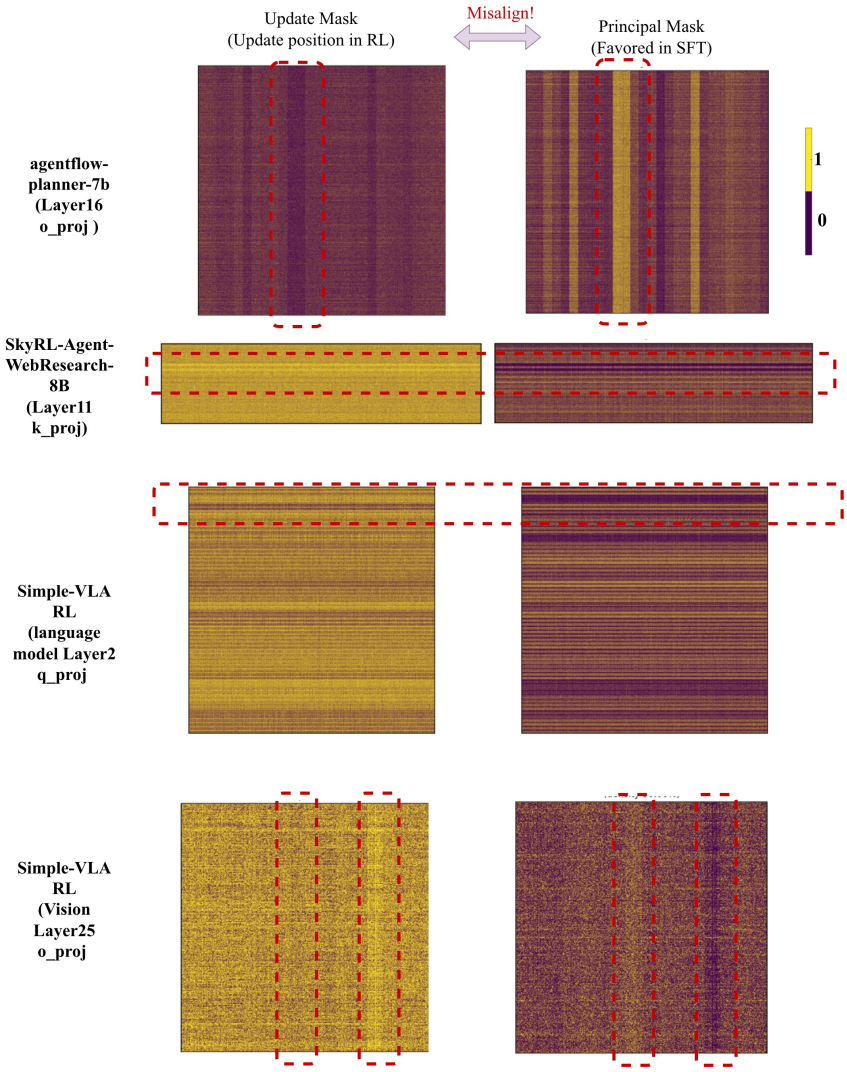

Figure 18: **Update–principal misalignment in RL-trained agents and embodied AI models.** The figure visualizes the bf16-aware *update mask* M_ℓ (left, showing locations changed under RL) versus the *principal mask* $P_\ell^{(k)}$ (right, showing a coordinate mask derived from the rank-k reconstruction) for representative layers across different domains. Top row: AGENTFLOW-PLANNER-7B (Layer 16, o_{proj}). Second row: SKYRL-AGENT-WEBRESEARCH-8B (Layer 11, k_{proj}). Third row: SIMPLE-VLA-RL (language model Layer 2, q_{proj}). Bottom row: SIMPLE-VLA-RL (Vision Layer 25, o_{proj}). Dashed red boxes highlight regions where RL updates concentrate *outside* principal-weight bands. This consistently shows that RL updates are misaligned with principal-weight coordinate masks, indicating a robust off-principal routing mechanism across agent, tool-use, and embodied AI settings.

F.7 Learning-Rate and Optimizer Artifact Check

A natural concern is that RLVR's stable spectra are merely an artifact of the small base learning rates commonly used with AdamW in stable RL training. We separate this learning-rate artifact hypothesis from the intrinsic RL update geometry with an optimizer stress test.

Stable RL pipelines typically require conservative effective policy motion; however, a small AdamW base learning rate does not by itself explain our observations. AdamW adapts effective step sizes per parameter, so a small scalar learning rate does not uniformly freeze parameter space. Moreover, a small learning rate in SFT may keep spectra unchanged simply because optimization stagnates, whereas RLVR can substantially improve performance while preserving spectral geometry.

To directly stress the learning-rate explanation, we train Qwen3-1.7B-Base with DAPO using vanilla SGD at a much larger learning rate, $\mathrm{lr} = 10^{-2}$, for 200 steps with train batch size 128. The run uses no explicit reference-KL penalty. We then measure singular-value drift and principal subspace rotation at the 10th layer.

Table 11: **High-learning-rate SGD stress test.** Even under vanilla SGD with $\mathrm{lr} = 10^{-2}$, spectral drift remains extremely small and top-k singular subspaces rotate by less than 0.1° in the measured layer. This argues against the explanation that RLVR's spectral stability is merely a small-Adam-learning-rate artifact.

Metric	k	`q_proj`	`gate_proj`
Mean relative change $\lvert\Delta\sigma/\sigma\rvert$	–	0.0284%	0.0001%
Mean absolute change $\lvert\Delta\sigma\rvert$	–	1.73×10^{-6}	1.78×10^{-6}
Maximum U rotation	128	0.0685°	0.0442°
Maximum U rotation	256	0.0791°	0.0560°
Maximum U rotation	512	0.0766°	0.0626°
Maximum V rotation	128	0.0485°	0.0713°
Maximum V rotation	256	0.0791°	0.0713°
Maximum V rotation	512	0.0791°	0.0862°

The result shows that even with a drastically larger SGD learning rate, the measured spectra and principal subspaces remain highly stable. Thus, the observed spectrum preservation is not explained by a trivial "small learning rate freezes the network" mechanism. Instead, it is consistent with the Gate-II picture: RLVR updates remain geometrically constrained and avoid disrupting KL-sensitive principal structure.

We do not report high-learning-rate AdamW runs because they collapse before a meaningful spectral comparison can be made; SGD provides a cleaner stress test of whether spectral stability requires the tiny AdamW base learning rate used in standard RLVR recipes.

F.8 LoRA results

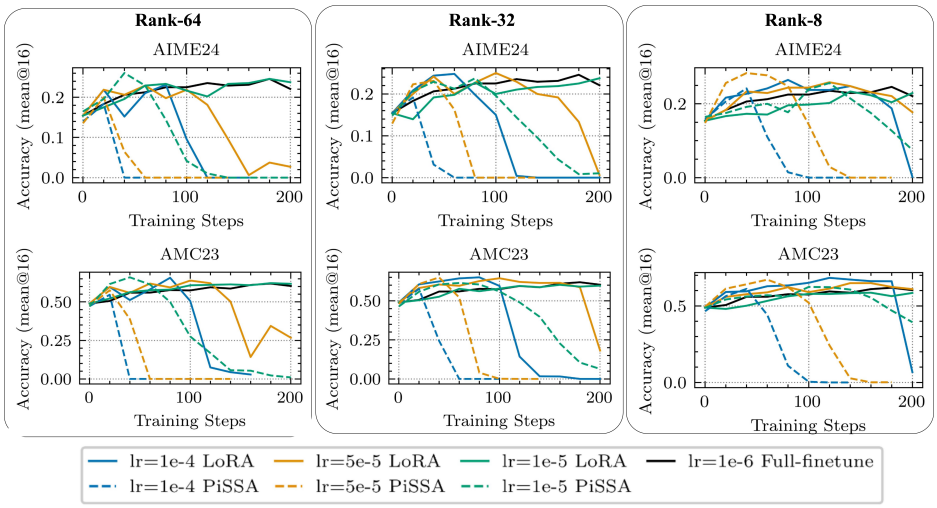

Figure 19: **LoRA vs. PiSSA on `DS-Qwen-1.5B` (DeepMath-103K).** We sweep ranks $\{8, 32, 64\}$ and learning rates $\{1\times10^{-4}, 5\times10^{-5}, 1\times10^{-5}\}$ for 200 steps, reporting pass@1 (avg@16) on AIME24 (top) and AMC23 (bottom). PiSSA initializes its trainable low-rank factors from the principal components. In these tested RLVR settings, this initialization provides *no consistent advantage* over LoRA and can become unstable and collapse at higher learning rates; LoRA remains more stable.

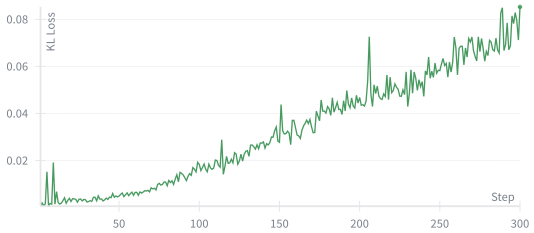

Figure 20: Token-wise KL loss. We show the token-wise KL loss during a DAPO run without a KL loss penalty, which shows a steadily increasing KL loss instead of being unconstrained.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the contributions of the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Discussed in the conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper provide complete proof for each theoretical result.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Experimental details needed for reproducibility are included both in the main draft and the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide our probe code in the supplementary.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Experimental setting for both training and evaluation is included in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For each evaluation, we generate N (N=12/16) rollouts per prompt and report the average.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Experiments compute resources needed are included in the Experimental Details in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our work adheres to the NeurIPS Code of Ethics. We do not use private or sensitive data, do not involve human subjects, and ensure transparency and reproducibility in our methodology and evaluations.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Pure research idea with no societal concern.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.

- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper will not release of data or models that have a high risk for misuse.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the code and data we use is allowed to use for research.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new assets are introduced in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLM is used only for writing, editing, or formatting purposes

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.