HOW MUCH, NOT WHERE: RLVR'S CAPABILITY LOSS IS DOSE-CONTROLLED AND DEEP-TAIL ONLY

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) reliably improves pass@1, yet can shrink the set of problems a model solves at large sampling budgets. We ask where that damage lives, when it becomes visible, and what controls it. Instrumenting GRPO training of Qwen3-1.7B-Base on MATH and MBPP, we track a forward-only signal: the per-prompt decay rate $\hat{\lambda}$ of the likelihood of frozen base-correct trajectories on *boundary* prompts, those the base model solves only rarely. We report three findings. **The damage is invisible at standard sampling depths.** Across our entire dose grid, pass@k stays significantly *above* base for every $k \leq 128$ and entropy settles on a floor rather than collapsing; harm appears only beyond $k \approx 192$, reaching -0.034 (95% CI $[-0.057, -0.010]$) at pass@256. Stopping at pass@64, the deepest endpoint in most of the inversion literature, would have shown no harm at all. **The damage is dose-controlled, and the dose is measurable early.** It grows monotonically with $\hat{\lambda}\times$steps; $\hat{\lambda}$ measured in the first 17% of training predicts the rate over the disjoint remainder ($r = 0.79$–0.94 on math) and grows superlinearly in learning rate. **Rehearsal removes the damage in proportion to dose, not targeting.** A dose- and schedule-matched control that anchors *randomly chosen* prompts protects as well as our signal-triggered scheme, and protection appears on prompts that were never anchored. The deliverable is a dose–response frontier trading deep-tail coverage against pass@1, not a per-prompt targeting rule. Code is available at `https://anonymous.4open.science/r/rlvr-dose`.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) is the standard recipe for improving the reasoning ability of large language models (Lambert et al., 2024; Guo et al., 2025; Shao et al., 2024; Kimi Team et al., 2025), and its headline metric, pass@1, improves reliably. But pass@1 is not capability. At large sampling budgets, the regime that governs repeated-sampling and test-time-scaling pipelines (Wang et al., 2023; Brown et al., 2024; Snell et al., 2024), RLVR-trained models are reported to solve *fewer* distinct problems than their base model, so the pass@k curves cross at large k (Yue et al., 2025; Zhou, 2026; Wu et al., 2025). Three questions remain open: at what sampling depth this inversion becomes visible, whether it can be anticipated before the training budget is spent, and whether it can be prevented.

Our starting point is a mechanistic picture we call *selection dynamics*. With G rollouts per prompt, a correct trajectory whose policy probability p satisfies $pG \ll 1$ is rarely sampled and so receives no positive gradient, while shared-parameter updates from other prompts squeeze it out: a multiplicative decay $\dot{p} \approx -\lambda p$ punctuated by rare rescue events (Williams, 1992; Mei et al., 2020; 2021). Because the decay is multiplicative, $\log p_t$ is near-linear early in training, so its rate can be measured early and cheaply. We track the teacher-forced likelihood of a frozen library of base-correct trajectories on *boundary* prompts (those the base model solves at rate $\in [10^{-3}, 10^{-1}]$) and fit a per-prompt slope $\hat{\lambda}$. This picture predicts that harm appears first at the largest sampling depths and grows with the dose $\hat{\lambda}\times$steps.

Where the damage lives. Likelihood squeeze-out is widespread at moderate dose, yet for hundreds of steps *no* capability damage appears at ordinary sampling depths. Policy entropy does not collapse to zero but settles on a floor, consistent with group-normalized GRPO gradients vanishing on

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

saturated groups, and pass@k stays significantly above base for every $k \leq 128$ across our whole dose grid, even as pass@1 grows by an order of magnitude. Only at $k \geq 192$ does the highest-dose arm turn negative, and only at $k = 256$ significantly so (-0.034 $[-0.057, -0.010]$). In our regime, the implicit assumption that moderate-k metrics reveal the harm is false: the same run delivers a $14\times$ pass@1 gain and a significant deep-tail loss.

What controls it. The damage is monotone in dose: across learning rates and horizons, pass@256 relative to base declines from significantly positive, through zero, to significantly negative. And $\hat{\lambda}$ supplies the dose in advance. Fitted on the first 50 steps, it predicts the squeeze-out rate over the disjoint remainder at $r = 0.79$–0.94 on math, and its median grows superlinearly in learning rate over the range we tested. A short probe therefore forecasts a configuration's risk before the budget is committed.

What removes it. Anchoring, which switches a prompt's loss from policy gradient to negative log-likelihood (NLL) on its stored base-correct trajectories, removes the deep-tail damage in proportion to how much of it is applied. *Selective dynamic anchoring* (SDA) applies it in closed loop, anchoring a prompt only while its rolling $\hat{\lambda}$ exceeds a threshold. In the damage regime, SDA restores pass@256 to parity with base while retaining three quarters of vanilla's pass@1 and touching about 15% of boundary prompts. But a control that replays SDA's anchoring schedule on *randomly chosen* prompts protects indistinguishably well, in both domains. Two analyses explain why. First, the protection is significant on prompts that were never anchored, so it propagates through shared parameters rather than memorizing anchored answers. Second, it scales with total anchoring dose: a dose ladder on code traces a monotone curve from partial to complete neutralization, at a monotonically rising pass@1 price. What $\hat{\lambda}$ buys is therefore the dose *trajectory*, that is, when to start, how much to hold, and when to release, not the identity of the prompts.

Contributions. (1) A characterization of *where* RLVR capability damage lives, in two domains: absent at $k \leq 128$, where policy entropy stabilizes on a floor, and emerging only at $k \gtrsim 192$, in proportion to the dose $\hat{\lambda}\times$steps. (2) $\hat{\lambda}$, a forward-only per-prompt telemetry signal costing 0.14% of training wall-clock, whose early estimate predicts the later squeeze-out rate and whose median grows superlinearly in learning rate. (3) A dose–response account of anchoring: protection depends on dose alone, since *which* prompts are anchored does not measurably matter and protection appears on prompts never anchored, together with the pass@1-versus-deep-tail frontier this implies for practice.

2 Related work

RLVR and GRPO. RLVR replaces RLHF's learned reward model (Christiano et al., 2017; Schulman et al., 2017; Ouyang et al., 2022) with programmatic verifiers (Lambert et al., 2024; Guo et al., 2025; Kimi Team et al., 2025), removing reward-model overoptimization (Gao et al., 2023) as the dominant failure mode. We work with GRPO (Shao et al., 2024), which replaces the value function with group-normalized advantages (Ahmadian et al., 2024; Yu et al., 2025; Liu et al., 2025b), and exploit one structural property of that normalization, zero gradient on saturated groups, to explain the entropy floor we observe.

pass@k inversion and anchoring. Yue et al. (2025) argue that RLVR mainly re-weights base capabilities and can shrink coverage at large k; Wu et al. (2025) formalize the support-preservation limit, while Liu et al. (2025a) and Wang et al. (2026b) report the opposite sign under prolonged training or sharper sampling primitives, and Zhai et al. (2026) extend the question to agentic PASS@(k, T). Related diagnostics track when RLVR cannot move a prompt at all (Chen et al., 2026) and when optimizing pass@k trades against pass@1 (Chen et al., 2025). Zhou (2026) localize the loss on boundary prompts and propose *static* per-problem base anchoring (PBA). Our results bear on this literature in two ways: the phenomenon is invisible below $k \approx 192$ at moderate dose, so endpoint depth is a confound across this line of work; and anchoring's protection is a function of dose rather than of per-problem assignment.

Entropy dynamics. Cui et al. (2025) derive the per-step covariance law $\Delta H \approx -\eta \operatorname{Cov}(\log \pi, A)$ and document entropy collapse as a central RLVR failure mode; follow-ups localize it to branching tokens (Wang et al., 2025), token position (Yang et al., 2026), on-policy entropy flow (Xu et al., 2026), and SFT overtraining upstream (Aphale & Liu, 2026), and diversity loss under preference-based RLHF has been documented in the same spirit (Kirk et al., 2024). Our observations refine this

picture: in group-normalized RLVR at 1.7B scale, entropy does not collapse to zero but reaches a self-stabilizing floor, consistent with moderate-k metrics staying healthy while the deep tail erodes.

Training dynamics and early signals. That RLVR training curves are low-dimensional and near-linear in the right coordinates has been reported for aggregate metrics (Wang et al., 2026a), echoing the tiny-subspace picture of gradient descent (Gur-Ari et al., 2018) and the finding that RL fine-tuning updates a small subnetwork (Mukherjee et al., 2025). $\hat{\lambda}$ is the per-prompt, forward-only counterpart, and it is what makes squeeze-out a *schedulable* quantity, a concern shared by work on *when* to apply RLVR pressure (Zhang et al., 2026). That the damage and its rescue propagate across prompts through shared parameters is consistent with the prompt-interference analysis of Barakat et al. (2026).

Forgetting, rehearsal, and policy-gradient theory. Anchoring is a rehearsal scheme in the sense of continual learning (McCloskey & Cohen, 1989; French, 1999; Robins, 1995; Kirkpatrick et al., 2017; Lopez-Paz & Ranzato, 2017), which remains relevant for LLM finetuning (Luo et al., 2023); SDA's distinguishing feature is that its *trigger* is a forward-only early-warning signal rather than a task boundary. Our dose-matched control is an ablation that rehearsal work rarely runs: does it matter *which* examples are rehearsed, beyond how much rehearsal happens? In both domains, our answer is that it does not (§5.3). The multiplicative-decay picture behind $\hat{\lambda}$ draws on convergence rates for softmax policy gradient, where progress constants scale with the initial probability of the optimal action (Mei et al., 2020; Agarwal et al., 2021), on the classical estimator (Williams, 1992; Sutton et al., 2000), and on committal behavior under finite sampling (Mei et al., 2021).

3 Selection dynamics and the squeeze-out rate $\hat{\lambda}$

3.1 Setup and the boundary stratum

Let π_θ be the policy, initialized at a base model π_0, and let $p_t^{(i)}$ denote the total probability mass π_{θ_t} assigns to verifier-accepted trajectories for prompt i. Before training we estimate $\hat{p}_0^{(i)}$ by large-scale base sampling ($n = 1024$ per prompt for MATH at T=1.0; $n = 1536$ for MBPP) and stratify the pool. *Boundary* prompts have $\hat{p}_0 \in [10^{-3}, 10^{-1}]$; for MBPP, which lacks a saturated stratum, we use any prompt with at least one base success, and §B re-runs the cross-domain comparison under MATH's definition. These are the prompts on which finite-sample selection can extinguish rare correct modes. The same sampling run yields a frozen *trajectory library* of up to 32 deduplicated base-correct trajectories per prompt, used for all telemetry and anchoring losses.

3.2 Telemetry and the squeeze-out rate

Every 10 optimizer steps we teacher-force the current policy on a fixed set of (boundary prompt, base-correct trajectory) pairs, 8 library trajectories per prompt, and record the mean per-token log-likelihood $\ell_t^{(i)}$. The pass is forward-only and adds 0.43 s to a 30.3 s step, i.e. 0.14% of training wall-clock (§C). The *squeeze-out rate* is the fitted early slope

$$\hat{\lambda}_i \;=\; -\,\mathrm{slope}_t\left[\ell_t^{(i)}\right] \qquad \text{(nats/token/step)}, \tag{1}$$

estimated over the first few telemetry points (offline analyses use the window up to step 50; the online trigger uses a rolling 5-point window).

What $\hat{\lambda}$ measures, and what it does not. $\hat{\lambda}$ is not a direct estimate of the rate λ in $\dot{p} \approx -\lambda p$. The two differ in units (λ is per step; $\hat{\lambda}$ is nats per token per step) and in object: $p_t^{(i)}$ is the total mass on *all* accepted trajectories, whereas $\ell_t^{(i)}$ is the mean log-likelihood of a *frozen, finite* set of base-correct ones. The gap has a concrete consequence. Because the policy can migrate onto *new* correct trajectories, a decline in ℓ does not by itself imply a decline in p; what $\hat{\lambda}$ measures is displacement away from the base model's solution modes. We use it as a proxy for squeeze-out pressure on rare modes and justify that use empirically rather than by derivation: cumulative $\hat{\lambda}\times$steps orders the deep-tail damage monotonically (§5.2), which is the property the signal is asked to deliver. The same section bounds the proxy: $\hat{\lambda}$ is distributional, not a per-prompt oracle.

3.3 ANCHORING AND ITS CONTROLS

Anchoring replaces a prompt's policy-gradient loss with an NLL loss on trajectories sampled from its frozen library. *Selective dynamic anchoring* (SDA) applies it in closed loop via a two-state machine per boundary prompt. All prompts start NORMAL (ordinary GRPO). When the rolling $\hat{\lambda}_i$ exceeds a threshold τ, prompt i becomes ANCHORED: it leaves the rollout pool and instead contributes an NLL loss on 2 sampled library trajectories (loss weight 1.0). When $\hat{\lambda}_i$ falls below $\tau - m$ (release margin m), it returns to NORMAL.

Threshold calibration. On MATH, τ was calibrated once from a pilot probe run at the target learning rate and not adjusted. On MBPP, that procedure failed: a threshold calibrated on whole-window slopes fires only during the early transient (§5.4). We therefore recalibrated on the online rolling estimator and then swept three thresholds. We report all three as a dose ladder rather than presenting the best one as a tuned configuration: the MBPP thresholds were chosen with knowledge of the outcome, and the ladder, not any single arm, is the result.

Controls. *Static PBA* anchors all boundary prompts for the whole run; it is our own reproduction of the scheme of Zhou (2026), not numbers taken from that work. *Pseudo-anchoring* replays a real SDA run's event stream (the same prompt count and the same anchor and release steps, hence the same total anchored prompt-steps) on randomly chosen prompts from the training pool, mapped one-to-one and disjoint from the triggered set. It matches dose and schedule exactly while destroying the targeting. On MATH the draw is over the whole pool, so 92 of the pseudo arm's 116 anchored prompts lie outside the boundary evaluation set; on MBPP all 66 lie inside it.

4 EXPERIMENTAL SETUP

Model and training. Qwen3-1.7B-Base (Yang et al., 2025), trained with GRPO: group size $G{=}8$, 256 prompts per step, temperature 1.0, at most 1024 new tokens, AdamW with $(\beta_1, \beta_2){=}(0.9, 0.95)$, gradient clipping, and DDP over 8 GPUs. Rollouts come from a colocated vLLM engine (Kwon et al., 2023) with prefix caching disabled; policy weights are synced in place into the engine every step, followed by a KV-cache reset, so rollouts are exactly on-policy. Zero-variance groups are excluded from the policy-gradient loss. **No arm applies a KL penalty to the base policy**, so we measure the unregularized squeeze-out dynamics; §6 discusses what this leaves untested. Full hyperparameters are in Table 5.

Domains and the evaluation set. MATH (Hendrycks et al., 2021): 3000-prompt training pool, 752 boundary evaluation prompts. MBPP (Austin et al., 2021): 970-prompt pool (one removed for length), 926 boundary evaluation prompts; rewards from sandboxed execution of the original test suites. **The evaluation prompts are a subset of the training pool.** All pass@k numbers therefore measure coverage on prompts the policy has trained on, not generalization; and because anchoring supervises base-correct trajectories on exactly these prompts, the anchored arms could in principle memorize evaluation answers. Section 5.3 tests that confound by splitting the evaluation set into anchored and never-anchored prompts; §6 states what remains untested.

Endpoints. Deep-tail coverage is pass@256 on boundary prompts, estimated from $n{=}256$ samples per prompt per checkpoint with the unbiased estimator of Chen et al. (2021), applied to both sides; base references are the $n{=}1024/1536$ base runs projected to $k{=}256$. The two sides are unbiased but not equally precise: at $k = n$ the RL-side estimate degenerates to "at least one success in 256 draws" and carries most of the paired noise. §B re-runs the headline comparisons against a base reference subsampled to $n{=}256$ and finds the conclusions unchanged. We report Δpass@256, the mean over boundary prompts of $\text{pass@256}_{\text{RL}} - \text{pass@256}_{\text{base}}$, with 95% CIs from a 10,000-resample bootstrap over prompts (Efron & Tibshirani, 1994); arm-versus-arm comparisons use a paired bootstrap on common prompts. These CIs cover prompt sampling only, not run-to-run variance (§6). Unless stated otherwise, all metrics are computed on the boundary evaluation set.

Dose grid. MATH: learning rates $\{4, 8, 12, 16\} \times 10^{-6}$, 300–600 steps. MBPP: $\{4, 8, 12\} \times 10^{-6}$, 150 steps.

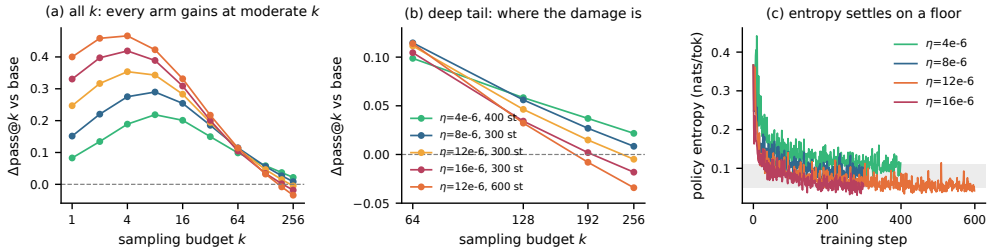

Figure 1: **The damage is deep-tail only. (a)** Δpass@k versus k for the MATH dose grid ($\log_2$ axis): every arm is far above base across the moderate-k range, and the arms order by dose. **(b)** The same curves for $k \geq 64$, with the y-axis rescaled to the deep tail: curves cross zero between $k = 128$ and $k = 192$, and only pass@256 separates arms into significantly negative territory. **(c)** Entropy at the four learning rates; the shared floor at 0.05–0.11 nats/token (shaded) is reached within $\sim$100 steps and held thereafter.

Table 1: **Deep-tail dose response (MATH).** All metrics on 752 boundary prompts (n=256 per prompt; base references from n=1024). Base references pass@1=0.031, pass@64=0.640, pass@256=0.860; CIs from bootstrap over prompts, 10k resamples. H is the mean policy entropy over the final 50 steps (nats/token). pass@64 is significantly above base on every arm; pass@256 versus base moves monotonically from significantly positive, through zero, to significantly negative, while pass@1 rises throughout. The monotone ordering refers to point estimates; adjacent-arm CIs overlap.

Arm ($\eta \times 10^6$ @ steps)	H	pass@1	Δpass@64 (95% CI)	Δpass@256 (95% CI)
base	—	0.031	—	—
4 @400	0.107	0.114	$+0.099$ $[+0.081, +0.118]$	$\mathbf{+0.022}$ $[+0.002, +0.042]$
8 @300	0.089	0.183	$+0.115$ $[+0.094, +0.136]$	$+0.008$ $[-0.014, +0.031]$
12 @300	0.067	0.278	$+0.112$ $[+0.089, +0.135]$	-0.005 $[-0.028, +0.018]$
16 @300	0.052	0.362	$+0.105$ $[+0.081, +0.129]$	-0.018 $[-0.043, +0.007]$
12 @600	0.053	0.431	$+0.114$ $[+0.091, +0.138]$	$\mathbf{-0.034}$ $[-0.057, -0.010]$

5 Results

5.1 At standard sampling depths, there is no damage—at any dose we tested

Boundary likelihood declines steadily under RLVR at every learning rate we ran, yet at ordinary sampling depths the trained policies are uniformly better than base. Across the dose grid of Table 1, pass@1 rises by up to $14\times$ and pass@64 sits significantly *above* its base reference of 0.640 on every arm, including the highest-dose arm, whose deep tail is already significantly damaged. Sweeping k (Figure 1a,b), *every* arm remains significantly above base at every $k \leq 128$; the top-dose arm first turns negative at $k = 192$ (-0.008, not significant) and is significantly negative only at $k = 256$. This is the signature of losing rare modes while keeping common ones. It also means that the sampling depth of the endpoint, not only the training dose, determines whether the phenomenon is observed at all.

Policy entropy is consistent with this picture: it falls quickly and then *stops*, settling at 0.05–0.11 nats/token (Figure 1c, Table 1) rather than collapsing toward zero. A mechanism consistent with both facts is visible in the loss: as groups saturate (all-correct or all-wrong), group-normalized advantages vanish and with them the sharpening gradient, so GRPO is self-limiting at this scale. We offer this as an account of an observed association, not a demonstrated cause, since we did not intervene on entropy. The methodological consequence is the point of this section: had we stopped at pass@64, the deepest endpoint in most of the inversion literature, we would have concluded that likelihood squeeze-out is a harmless indicator of ordinary distribution shift.

Table 2: **Squeeze-out telemetry.** Median per-prompt $\hat{\lambda}$ over the full run, and the correlation between the early-window estimate (steps $\leq$50) and the rate over the *disjoint* remainder. The nested column (early window against a full window that contains it) is reported for transparency because it is the quantity a naive analysis produces; all claims in the text use the disjoint column. MBPP horizons are 150 steps, MATH 300.

Domain	η	n	median $\hat{\lambda}$ (nats/tok/step)	r (disjoint)	r (nested)
MATH	4×10^{-6}	735	1.07×10^{-4}	0.942	0.966
MATH	8×10^{-6}	735	3.17×10^{-4}	0.903	0.932
MATH	12×10^{-6}	735	6.18×10^{-4}	0.803	0.860
MATH	16×10^{-6}	735	1.10×10^{-3}	0.791	0.851
MBPP	4×10^{-6}	863	6.49×10^{-3}	0.734	0.869
MBPP	8×10^{-6}	863	1.32×10^{-2}	0.738	0.915
MBPP	12×10^{-6}	863	1.41×10^{-2}	0.354	0.941

5.2 The damage is dose-controlled, and $\hat{\lambda}$ measures the dose early

Table 1 reads as a dose–response curve: as the product of learning rate and step count grows, Δpass@256 declines monotonically from significantly positive to significantly negative. The 600-step run at $\eta = 1.2 \times 10^{-5}$ is the clearest case: Δpass@256 falls from -0.005 at step 300 to -0.034 $[-0.057, -0.010]$ at step 600 while pass@1 rises from 0.031 to 0.431, a $14\times$ pass@1 gain and a significant deep-tail loss in one run. A second seed reproduces the endpoint (-0.028 $[-0.052, -0.003]$, pass@1 0.438).

A 50-step probe, 17% of the MATH training horizon, is enough to rank prompts by the rate at which they will be squeezed out over the rest of the run (Table 2, Figure 2a). Because the early and late windows are disjoint, this is a prediction rather than a restatement. The correlation degrades gracefully as the dynamics accelerate but stays at $r \geq 0.79$ across the MATH grid. On code the probe is weaker and degrades sharply with learning rate, from $r = 0.73$ to $r = 0.35$: where the dynamics are fastest, an early window buys the least.

The rate itself is disproportionately a high-learning-rate phenomenon. On math, median $\hat{\lambda}$ grows superlinearly over the range we tested, roughly as $\eta^{1.66}$ (log-log fit over 4 points, $R^2 = 0.997$; Figure 2b), so halving the learning rate cuts the squeeze-out dose by more than a factor of three. This is what makes a probe actionable rather than merely descriptive. Code differs in both magnitude and shape: $\hat{\lambda}$ is an order of magnitude larger and *saturates* rather than accelerating, growing as $\eta^{0.74}$ over three points and flattening between 8 and 12×10^{-6}. With three points, two of which nearly coincide, we describe this as a transition from growth to saturation rather than fitting a scaling law; §5.4 shows that the deep-tail damage plateaus over the same interval.

Limits of the signal. $\hat{\lambda}$ is distributional, not a per-prompt oracle. Within the boundary stratum it is essentially uncorrelated with initial difficulty $\hat{p}_0$ ($r = 0.008$, $n = 735$; §B), so it is not a re-description of how hard a problem is. But it also does not identify *which* prompt will be lost: against per-prompt Δpass@256 it correlates at $r = -0.065$, and the prompts that go extinct decay no faster than the population; they simply start lower (Figure 5a). Extinction is probabilistic at the individual level, as the multiplicative picture predicts, and our claims are correspondingly distributional. Section 5.3 shows the interventional counterpart of this fact.

5.3 Rehearsal removes the damage in proportion to dose, not to targeting

Anchoring 15% of boundary prompts, and only while they are being squeezed out, is enough to erase the deep-tail loss (Table 3, upper block). SDA ($\tau = 1.85 \times 10^{-3}$, margin 10^{-3}) anchors 116 of 752 boundary prompts in a trigger–release loop spanning the run (steps 21 to 591; rasters in Figure 4). It restores pass@256 to statistical parity with base and keeps three quarters of vanilla's pass@1 (0.325 vs. 0.431). The protection is not transient: the paired advantage grows from $+0.024$ at step 300 to $+0.037$ at step 600 as vanilla's damage compounds. Static PBA, which anchors every

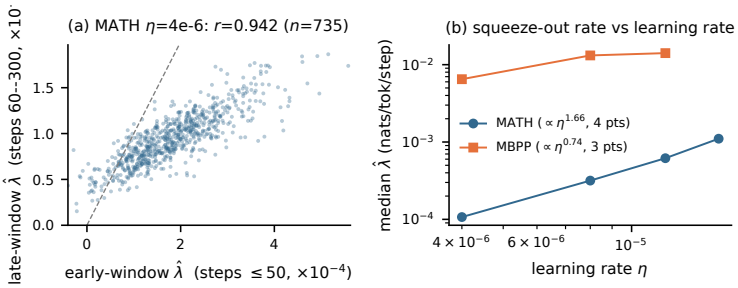

Figure 2: **Squeeze-out telemetry. (a)** Per-prompt $\hat{\lambda}$ on steps $\leq$50 versus $\hat{\lambda}$ on the disjoint window (steps 60–300), MATH η=4×10^{-6}; axes clipped at the 99.5th percentile, dashed diagonal. Points sit below the diagonal because squeeze-out is fastest early: the early estimate ranks prompts well but is biased upward as a level. **(b)** Median $\hat{\lambda}$ versus learning rate, log-log. Math grows superlinearly; code is an order of magnitude larger but saturates over the top of the range, mirroring the damage plateau of §5.4.

Table 3: **Anchoring in the damage regime (MATH, η=1.2×10^{-5}, 600 steps, 752 boundary prompts).** Paired Δ is a paired bootstrap of arm$-$vanilla on common prompts. The lower block splits the evaluation set by whether SDA anchored the prompt; on those subsets vanilla itself is damaged by -0.123 $[-0.192, -0.056]$ (116 anchored) and -0.018 $[-0.043, +0.006]$ (636 never anchored). Pseudo protects indistinguishably from SDA, and both protect significantly on prompts they never anchored; static PBA anchors every evaluation prompt, so its number alone is not separable from memorization.

Arm	step	pass@1	Δpass@256 (95% CI)	Paired Δ vs. vanilla
vanilla	300	0.278	-0.005 $[-0.028, +0.018]$	—
vanilla	600	0.431	$\mathbf{-0.034}$ $[-0.057, -0.010]$	—
SDA	300	0.203	$+0.019$ $[-0.004, +0.041]$	$+0.024$ $[+0.000, +0.048]$
SDA	600	0.325	$+0.003$ $[-0.019, +0.026]$	$\mathbf{+0.037}$ $[+0.015, +0.060]$
pseudo	600	0.308	-0.004 $[-0.026, +0.019]$	$\mathbf{+0.031}$ $[+0.008, +0.053]$
static PBA	600	0.123	$\mathbf{+0.075}$ $[+0.058, +0.092]$	$\mathbf{+0.109}$ $[+0.082, +0.134]$
			paired Δ at step 600, split by anchoring	
			116 anchored by SDA	636 never anchored
SDA	600		$\mathbf{+0.086}$ $[+0.017, +0.155]$	$\mathbf{+0.028}$ $[+0.005, +0.052]$
pseudo	600		$+0.035$ $[-0.035, +0.103]$	$\mathbf{+0.030}$ $[+0.006, +0.054]$

boundary prompt throughout, pushes the deep tail well above base ($+0.075$) but forfeits most of RLVR's benefit, with pass@1 at 0.123.

Targeting contributes nothing measurable. The pseudo arm replays SDA's event stream (anchor and release steps and counts, hence the same 1788 anchored prompt-steps) on randomly drawn prompts disjoint from the triggered set. It neutralizes the damage as well as SDA does, at essentially the same pass@1 cost, in both domains: SDA$-$pseudo is $+0.007$ $[-0.013, +0.027]$ on MATH and $+0.001$ $[-0.010, +0.012]$ on MBPP. Neither is significant, and the MATH effect equals the difference between two seeds of the identical vanilla configuration (0.007). We therefore state a bounded null rather than an equivalence: our design can exclude a targeting benefit larger than about 0.02 Δpass@256, roughly half the total protection on offer, and cannot resolve anything smaller (§6).

The protection is not memorization of anchored answers. Because evaluation prompts are training prompts and anchoring supervises base-correct trajectories on them, the natural alternative explanation is that anchored arms memorize the answers they are shown. The lower block of Table 3 tests this by splitting the evaluation set. On the 636 boundary prompts SDA never anchored, vanilla sits at -0.018 and SDA at $+0.010$, a significant paired protection of $+0.028$ $[+0.005, +0.052]$; pseudo, whose anchored set is disjoint from SDA's, delivers the same $+0.030$ there. The code replication is cleaner still: the 66 prompts sda2 anchored have base pass@256 $\approx$ 1.0 and hence no headroom to

Table 4: **Code replication and the anchoring dose ladder (MBPP, η=8×10^{-6}, 150 steps; 926 boundary prompts; base $\mathbb{E}$[pass@256]=0.946, pass@1=0.199).** The anchoring arms differ only in dose, given as anchored prompt-steps: sda1 (whole-window threshold, 24 prompts), sda2 (rolling threshold 1.8×10^{-2}, 66 prompts), pseudo (sda2's schedule on 66 random prompts), sda3 (threshold 1.4×10^{-2}, 148 prompts). Protection is monotone in dose; targeting is immaterial.

Arm	dose	step	pass@1	Δpass@256 (95% CI)	Paired Δ vs. vanilla
vanilla	0	75	0.785	-0.039 [$-0.055, -0.024$]	—
vanilla	0	150	0.857	$\mathbf{-0.051}$ [$-0.068, -0.035$]	—
vanilla, η=4×10^{-6}	0	150	0.708	-0.021 [$-0.035, -0.006$]	—
vanilla, η=12×10^{-6}	0	150	0.890	-0.040 [$-0.056, -0.025$]	—
sda1	260	150	0.850	-0.039 [$-0.054, -0.024$]	$+0.012$ [$-0.002, +0.026$]
sda2	759	75	0.661	$+0.011$ [$-0.000, +0.022$]	$\mathbf{+0.050}$ [$+0.034, +0.066$]
sda2	759	150	0.809	-0.016 [$-0.028, -0.004$]	$\mathbf{+0.035}$ [$+0.018, +0.050$]
pseudo (random)	759	150	0.826	-0.017 [$-0.031, -0.004$]	$\mathbf{+0.034}$ [$+0.018, +0.049$]
sda3	1870	150	0.603	$\mathbf{+0.019}$ [$+0.010, +0.029$]	$\mathbf{+0.070}$ [$+0.052, +0.089$]

memorize into, so the entire measured protection lives on the 860 prompts it never anchored ($+0.037$ [$+0.021, +0.055$]). On MATH, 92 of the pseudo arm's 116 anchored prompts are not in the evaluation set at all. Anchoring's benefit therefore propagates through shared parameters to prompts it never touches. The one arm this argument does not rescue is static PBA: its $+0.075$ cannot be separated from memorization by any split of this set, and we do not read it as capability preservation.

What the signal supplies is the dose trajectory. These results relocate $\hat{\lambda}$'s interventional role. It does not tell us *which* prompts to protect, consistent with §5.2, but it does supply the schedule: when anchoring must begin, how much to hold, and when release is safe. That trajectory is exactly what the pseudo arm had to copy from a real SDA run in order to exist at all.

5.4 Code fails four times faster, and traces the full dose–response curve

MBPP reproduces the whole phenomenology with the dial turned up (Table 4). At matched learning rate, code-domain $\hat{\lambda}$ is an order of magnitude larger than math's: $30\times$ when MBPP is re-stratified to MATH's $\hat{p}_0$ criterion, $42\times$ under each domain's own boundary definition (§B); we quote the matched figure. Training for 150 steps at $\eta = 8\times10^{-6}$ produces deep-tail damage of -0.051 [$-0.068, -0.035$], which math reaches only after 600 steps at $\eta = 1.2\times10^{-5}$, so code fails roughly four times faster. Damage is again dose-dependent in steps (-0.039 at step 75) and in learning rate (-0.021 at η=4×10^{-6}), and again coexists with a large pass@1 gain ($0.199 \to 0.857$). Entropy again settles on a floor, including at $\eta = 1.2\times10^{-5}$, where training remains stable (§B), so the floor is a property of group-normalized GRPO rather than of the math domain.

Two features are specific to code. First, the damage *plateaus*: at η=12×10^{-6} it reaches -0.040, statistically indistinguishable from the -0.051 of the η=8×10^{-6} arm despite the higher rate, and it is near-saturated by step 75. This mirrors the saturation of $\hat{\lambda}$ over the same interval (§5.2) and is consistent with the boundary stratum's squeezable mass being largely exhausted at these rates. Second, the calibration failure of §3.3: a threshold set from whole-window slopes fires only during the early transient, so sda1 anchors 24 prompts during steps 31–51, protects the midpoint, and has faded by step 150 ($+0.012$ paired, not significant).

The resulting four-arm ladder is the most informative object in the paper (Figure 3c). Paired protection rises monotonically with dose: $+0.012$ at 260 anchored prompt-steps, $+0.035$ at 759 (and $+0.034$ for the same dose spent on random prompts), and $+0.070$ at 1870, while the identity of the anchored prompts stays immaterial. At the top, sda3 *over*-neutralizes, lifting pass@256 above base ($+0.019$) but retaining only 70% of pass@1; in between, sda2 removes 68% [$46\%, 90\%$] of the endpoint damage ($-0.051 \to -0.016$) at a 6% pass@1 cost. The deliverable is the curve, not any single arm.

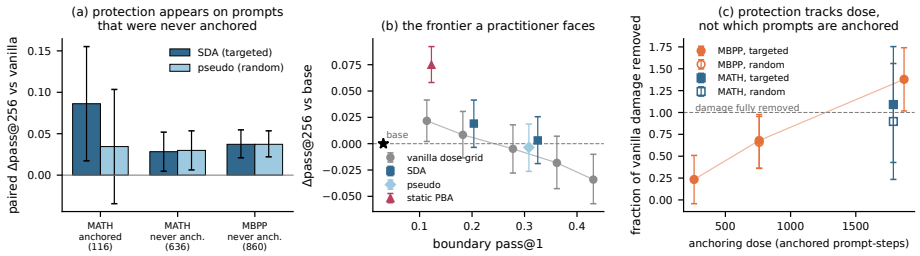

Figure 3: **Anchoring protects by dose, and the practitioner's frontier.** **(a)** Paired Δpass@256 versus vanilla, split by whether prompts were anchored; the MBPP anchored subset is omitted because its base pass@256 is ≈ 1.0, leaving no headroom. **(b)** Δpass@256 against pass@1 for every MATH arm; the vanilla dose grid (grey) already traces a frontier that lowering the dose can ride. **(c)** Protection normalized by each domain's vanilla damage, against dose in anchored prompt-steps; open markers are the dose-matched random controls.

6 Discussion and Limitations

What anchoring buys over simply training less. The cheapest intervention against a dose-controlled effect is a smaller dose, and Figure 3b shows that the vanilla grid alone traces a usable pass@1/deep-tail frontier. The comparison anchoring must beat is therefore specific. Vanilla stopped at step 300 sits at pass@1 0.278 with $\Delta\text{pass@256} = -0.005$, already at statistical zero, while SDA at step 600 reaches 0.325 at $+0.003$: a net $+0.047$ pass@1 at equal deep-tail safety. That gain is real, but far smaller than a paired comparison against a full-length vanilla run suggests. On code the gap is larger: sda2 at step 150 holds pass@1 at 0.809 with a tail of -0.016, whereas the closest vanilla point ($\eta{=}4{\times}10^{-6}$) reaches only 0.708 at -0.021. Anchoring's value is not that it rescues capability no cheaper method can, but that it decouples the two axes.

Practical recipe. Probe for about 20% of the budget and measure $\hat{\lambda}$; it forecasts the dose a configuration will deliver, not which prompts will be lost. If the projected dose is in the damage regime, first lower η or the step count, since superlinearity means small reductions buy a lot, and turn to anchoring when the pass@1 that dose would have bought is worth keeping. Calibrate the threshold on the same rolling estimator used online, and treat it as the dose knob it is.

Limitations. (i) *Evaluation prompts are training prompts.* All pass@k results measure coverage on the training pool, not generalization. The anchored/never-anchored split (§5.3) rules out memorization as the explanation of anchoring's benefit for SDA and pseudo, though not for static PBA, but converts none of our numbers into generalization claims. (ii) *Missing baselines.* No arm uses a KL penalty to the base policy, the standard defense against exactly this drift (Schulman et al., 2017; Ouyang et al., 2022); we can claim only that anchoring works without KL, not that it beats it. A *constant* random-rehearsal schedule would separately test whether the closed loop is needed to place the dose in time or only to size it; static PBA is our one signal-free point, at the curve's far end. (iii) *Single seed for the intervention arms.* Only vanilla MATH was run twice. Our CIs bootstrap over prompts and exclude run-to-run variance, which for that configuration was 0.007 Δpass@256, the same magnitude as the SDA$-$pseudo difference we report as a bounded null. (iv) *Post-hoc thresholds on code.* The three MBPP thresholds were chosen after the first calibration failed; we present them as a dose ladder, not a tuned method. (v) *Scale, scope, and magnitude.* One model scale (1.7B), one algorithm, two domains; the location of the damage threshold surely moves with scale, though we expect the mechanism to survive. The absolute damage is 0.03–0.05 pass@256: what concerns us is its monotone growth with dose and its invisibility to standard endpoints, not its size today.

AI USE STATEMENT

In this work, we did not use generative AI tools for any task with required disclosure: generative AI was not used to generate synthetic data, develop theoretical models or conceptual frameworks, formulate mathematical claims or assist in proving them, propose or refine hypotheses, design or provide feedback on the research methodology or experiments, implement methods, clean or reformat datasets, or interpret results; translation and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for two tasks with recommended disclosure: editing the paper to improve readability, and reviewing author-written code. We have reviewed all AI-assisted work: every edited passage was checked by the authors against the underlying results and revised where the suggested wording changed the meaning, and code-review comments were treated as suggestions that the authors evaluated and applied by hand, with the final code tested and verified by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All training, telemetry, evaluation, and analysis code, all run configs (every run in this paper is a JSON config), the frozen trajectory libraries, base-sampling statistics, per-run telemetry parquets, and anchoring event streams are provided as supplementary material; code is available at `https://anonymous.4open.science/r/rlvr-dose`. Deep-tail endpoints use fixed seeds and the unbiased pass@k estimator; every table cell can be regenerated from the released verdict files by a single analysis script.

REFERENCES

Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021. URL https://arxiv.org/abs/1908.00261.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, 2024. URL https://arxiv.org/abs/2402.14740. arXiv:2402.14740.

Siddharth Aphale and Kelly Liu. SFT overtraining predicts rank inversion via entropy collapse under RLVR. *arXiv preprint arXiv:2606.18487*, 2026. URL https://arxiv.org/abs/2606.18487. Deep Learning for Code (DL4C) Workshop at ICML 2026.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021. URL https://arxiv.org/abs/2108.07732.

Anas Barakat, Souradip Chakraborty, Khushbu Pahwa, and Amrit Singh Bedi. Why pass@k optimization can degrade pass@1: Prompt interference in LLM post-training. *arXiv preprint arXiv:2602.21189*, 2026. URL https://arxiv.org/abs/2602.21189.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024. URL https://arxiv.org/abs/2407.21787.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021. URL https://arxiv.org/abs/2107.03374.

Yulin Chen, He He, and Chen Zhao. The unlearnability phenomenon in RLVR for language models. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2605.16787. arXiv:2605.16787.

Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@k training for adaptively balancing exploration and exploitation of large reasoning models. *arXiv preprint arXiv:2508.10751*, 2025. URL https://arxiv.org/abs/2508.10751.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.03741. arXiv:1706.03741.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL https://arxiv.org/abs/2110.14168.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025. URL https://arxiv.org/abs/2505.22617.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall/CRC, 1994.

Robert M. French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, 3(4):128–135, 1999.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023. URL https://arxiv.org/abs/2210.10760. arXiv:2210.10760.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Nature*, 645:633–638, 2025. URL https://arxiv.org/abs/2501.12948. arXiv:2501.12948.

Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018. URL https://arxiv.org/abs/1812.04754.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2021. URL https://arxiv.org/abs/2103.03874. arXiv:2103.03874.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, et al. Kimi k1.5: Scaling reinforcement learning with LLMs. *arXiv preprint arXiv:2501.12599*, 2025. URL https://arxiv.org/abs/2501.12599.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.06452. arXiv:2310.06452.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024. URL https://arxiv.org/abs/2411.15124.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2305.01210. arXiv:2305.01210.

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

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025b. URL https://arxiv.org/abs/2503.20783.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://arxiv.org/abs/1706.08840. arXiv:1706.08840.

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

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pp. 109–165. Academic Press, 1989.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvári, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, pp. 6820–6829. PMLR, 2020.

Jincheng Mei, Bo Dai, Chenjun Xiao, Csaba Szepesvári, and Dale Schuurmans. Understanding the effect of stochasticity in policy optimization. In *Advances in Neural Information Processing Systems*, volume 34, pp. 19339–19351, 2021. URL https://arxiv.org/abs/2110.15572. arXiv:2110.15572.

Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tür, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2505.11711. arXiv:2505.11711.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022. URL https://arxiv.org/abs/2203.02155.

Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2): 123–146, 1995.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024. URL https://arxiv.org/abs/2408.03314.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, volume 12, pp. 1057–1063, 2000.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2506.01939. arXiv:2506.01939.

Tianle Wang, Jiayu Liu, Zhongyuan Wu, Shenghao Jin, Wei Chen, Hao Xu, and Ning Miao. Linear dynamics in the RLVR training of large language models. *arXiv preprint arXiv:2601.04537*, 2026a. URL https://arxiv.org/abs/2601.04537.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2203.11171. arXiv:2203.11171.

Zhilin Wang, Yafu Li, Shunkai Zhang, Zhi Wang, Haoran Zhang, Xiaoye Qu, and Yu Cheng. New skills or sharper primitives? a probabilistic perspective on the emergence of reasoning in RLVR. *arXiv preprint arXiv:2602.08281*, 2026b. URL https://arxiv.org/abs/2602.08281.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992.

Fang Wu, Weihao Xuan, Ximing Lu, Mingjie Liu, Yi Dong, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why RLVR may or may not escape its origin. *arXiv preprint arXiv:2507.14843*, 2025. URL https://arxiv.org/abs/2507.14843.

Huimin Xu, Shuai Zhao, Xiaobao Wu, and Anh Tuan Luu. Understanding and preventing entropy collapse in RLVR with on-policy entropy flow optimization. *arXiv preprint arXiv:2605.11491*, 2026. URL https://arxiv.org/abs/2605.11491.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Shumeng Yang, Yisu Liu, Jiayi Zheng, Zhaohui Yang, and Linjing Li. PAEC: Position-aware entropy calibration for LLM reasoning in RLVR. *arXiv preprint arXiv:2606.08543*, 2026. URL https://arxiv.org/abs/2606.08543.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025. URL https://arxiv.org/abs/2503.14476.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2504.13837. arXiv:2504.13837.

Zhiyuan Zhai, Wenjing Yan, Xiaodan Shao, and Xin Wang. Does RL expand the capability boundary of LLM agents? a PASS@(k,T) analysis. *arXiv preprint arXiv:2604.14877*, 2026. URL https://arxiv.org/abs/2604.14877.

Jinghao Zhang, Ruilin Li, Feng Zhao, and Jiaqi Wang. Not only where, but when: Temporal scheduling for RLVR. *arXiv preprint arXiv:2605.25381*, 2026. URL https://arxiv.org/abs/2605.25381.

Todd Zhou. When RLVR shrinks the reasoning boundary: Diagnosing pass@k inversion. *arXiv preprint arXiv:2607.20543*, 2026. URL https://arxiv.org/abs/2607.20543.

Table 5: Training hyperparameters (both domains unless noted).

Model	Qwen3-1.7B-Base (tied embeddings)
Algorithm	GRPO, $G = 8$, group-normalized advantages, no KL penalty
Prompts per step	256 (DDP over 8 ranks)
Optimizer	AdamW, $\beta = (0.9, 0.95)$, warmup 10 steps, grad clip 1.0
Learning rates	MATH $\{4, 8, 12, 16\} \times 10^{-6}$; MBPP $\{4, 8, 12\} \times 10^{-6}$
Sampling (train)	$T = 1.0$, top-p 1.0, max 1024 new tokens
Micro-batching	token budget 8192 per micro-batch, bsz 16
Rollout engine	colocated vLLM, mem fraction 0.35, eager mode, prefix caching off, per-step in-place weight sync + KV-cache reset
Telemetry	every 10 steps; 8 trajectories/prompt, teacher-forced
SDA (MATH, η=12×10^{-6})	$\tau = 1.85 \times 10^{-3}$ (pilot-probe calibration), margin 10^{-3}
SDA (MBPP, η=8×10^{-6})	sda1 τ=3.4×10^{-2}; sda2 τ=1.8×10^{-2}, margin 10^{-2}; sda3 τ=1.4×10^{-2}
Anchoring loss	NLL on 2 library trajectories/prompt, weight 1.0
Deep-tail eval	$n = 256$/prompt, $T = 1.0$, seed 1; MBPP max 512 tokens, stop `[DONE]`

A Hyperparameters and additional setup detail

Verifiers. MATH: answer extraction and equivalence checking, following the final-answer verification convention of Cobbe et al. (2021). MBPP: sandboxed execution against the original test suites (pass = all tests green); we deliberately keep the original suites rather than the strengthened ones of Liu et al. (2023), so that base and RL policies are scored by exactly the verifier used during training. The trajectory library keeps only verified-correct base trajectories (23,596 trajectories over 863 prompts for MBPP; 80,706 over 2,836 prompts for MATH, covering 735 of the 752 boundary prompts; the remaining 17 have only truncated correct trajectories and are excluded from telemetry, which is why telemetry $n = 735$).

Base references. Unbiased pass@k projection $1 - \binom{n-c}{k} / \binom{n}{k}$ (Chen et al., 2021) from base runs with $n = 1024$ (MATH) / $n = 1536$ (MBPP) per prompt.

B Additional analyses

Discriminant checks on $\hat{\lambda}$. Within the boundary stratum, $\hat{\lambda}$ is essentially uncorrelated with initial difficulty: $r = 0.008$ against $\hat{p}_0$ ($n = 735$), with $\hat{p}_0$-quartile medians of $6.03/5.92/6.19/6.52 \times 10^{-4}$ at η=1.2×10^{-5}. It is likewise uninformative about individual extinction: against per-prompt Δpass@256 it correlates at $r = -0.065$, and the 77 extinguished prompts have a median $\hat{\lambda}$ of 6.31×10^{-4} against a population median of 6.17×10^{-4}. Figure 5a shows the same fact in trajectory form: prompts that go extinct sit about 0.29 nats/token below the population at initialization and decline by an almost identical amount over 600 steps (0.35 versus 0.34 nats/token). What determines extinction is where a prompt starts, not how fast it falls.

Base down-sampling check. The RL side of every Δpass@256 is estimated from $n = k = 256$, where the estimator degenerates to an indicator, while the base side is projected from $n = 1024$. To confirm that the asymmetry does not drive the result, we re-drew base references by hypergeometric subsampling to $n = 256$ (20 replicates). The headline endpoints are unchanged: vanilla@600 gives -0.034 against the full reference and -0.034 ± 0.011 against the subsampled one; SDA@600 gives $+0.003$ and $+0.004 \pm 0.011$.

MBPP under the MATH boundary definition. MBPP's boundary set (any prompt with ≥ 1 base success) is broader than MATH's ($\hat{p}_0 \in [10^{-3}, 10^{-1}]$) and includes many prompts the base model solves easily: its mean $\hat{p}_0$ is 0.199, far above MATH's boundary range. Applying MATH's criterion to MBPP leaves 337 prompts with mean $\hat{p}_0 = 0.033$. On this matched stratum the code phenomenology is *stronger*, not weaker: vanilla@150 damage is -0.142 $[-0.184, -0.102]$ (against -0.051 on the full boundary set), sda2 -0.044, pseudo -0.053, sda3 $+0.039$. Median $\hat{\lambda}$ on the matched stratum is 9.56×10^{-3}, so the cross-domain ratio at η=8 × 10^{-6} is 30× rather than 42×. Both the order-

ing of arms and the "code is the fragile domain" conclusion survive the re-stratification; the exact multiplier does not, which is why §5.4 quotes the matched figure.

Code-domain threshold calibration. Rolling 5-point $\hat{\lambda}$ on the MBPP $\eta{=}8{\times}10^{-6}$ probe peaks at steps 50–60 (p90 ≈ 3.6–4.1×10^{-2}) and relaxes to 1–2.5×10^{-2} afterwards; a threshold calibrated on whole-window slopes (3.4×10^{-2}) is exceeded only near the peak, which explains sda1's 24-prompt, 20-step anchoring burst. Matching the calibration estimator to the online estimator (sda2, $\tau{=}1.8 \times 10^{-2}$) sustains anchoring; its event stream (76 anchor / 75 release over 66 prompts) shows the same anchor→recover→release micro-dynamics as MATH (142 anchor / 140 release over 116 prompts). Figure 4 shows all three rasters.

Code training stability at high learning rate. The $\eta{=}1.2{\times}10^{-5}$ MBPP run trains without incident (terminal reward 0.93, no gradient anomalies, entropy settling on the same floor as the lower rates), which is why we read the deep-tail plateau of §5.4 as saturation of the squeezable mass rather than as instability.

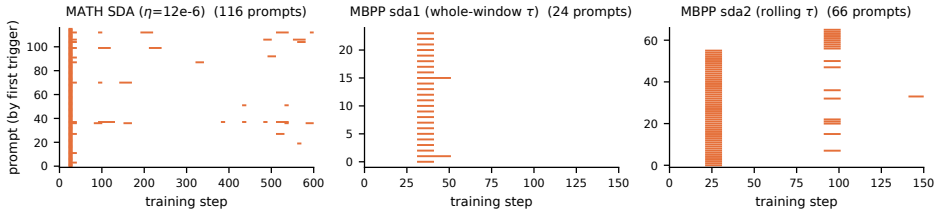

Figure 4: **Anchoring trigger dynamics.** Event rasters (prompt × step, rows sorted by first trigger). Left: MATH SDA at $\eta{=}1.2{\times}10^{-5}$ (116 prompts; a dense early wave plus sporadic triggers across the full run). Center: MBPP sda1, threshold calibrated on whole-window slopes (24 prompts, confined to steps 31–51). Right: MBPP sda2, rolling-estimator threshold (66 prompts; waves at steps 21–30 and ∼90–100 plus sporadic events). The threshold determines the temporal coverage of anchoring, which is what sets the dose.

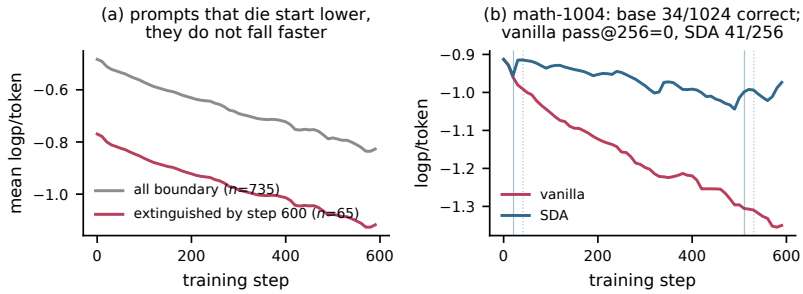

Figure 5: **Squeeze-out at the prompt level (MATH, $\eta{=}1.2{\times}10^{-5}$). (a)** Mean log-likelihood per token of frozen base-correct trajectories, for all boundary prompts versus the 66 prompts whose pass@256 falls to zero by step 600 (restricted to prompts with ≥ 4 base successes, so the loss is not a sampling artifact). The extinguished set starts lower and declines in parallel, consistent with extinction being driven by initial position rather than by an elevated rate. **(b)** A single prompt: the base model solves it 34 times in 1024 draws. Under vanilla the likelihood of its base solutions falls steadily and the RL policy solves it 0 times in 256; under SDA (vertical lines mark anchor and release events) the likelihood is held and the policy solves it 41 times in 256. 35 of the 66 extinguished prompts are recovered this way.

C Compute Cost

Telemetry overhead is measured directly by comparing steps that carry a telemetry pass against those that do not within the same run: 30.73 s versus 30.30 s mean step time on the MATH $\eta{=}1.2 \times 10^{-5}$

Table 6: Measured compute for the experiments reported here (8×H20 per job). Training figures are summed per-step wall-clock × 8 ranks over the 15 arms in the paper; evaluation figures are wall-clock spans of the deep-tail sampling jobs over 30 checkpoint evaluations. Excludes exploratory and discarded runs. The telemetry share here is telemetry GPU-hours summed over all 15 arms divided by total training GPU-hours; the 0.14% quoted in the text is the per-step measurement on a single MATH run (below). The two differ only in denominator, since mean step time varies across arms and domains.

Stage	GPU-hours	share
Base sampling, stratification, trajectory libraries	107	17%
RLVR training, 15 arms (MATH 8 arms, 292 h; MBPP 7 arms, 90 h)	382	63%
of which: $\hat{\lambda}$ telemetry overhead	0.3	0.08% of train
Deep-tail evaluation, 30 checkpoints, 256 samples/prompt	124	20%
Total	613	

run, i.e. 0.43 s per telemetry point, which amortized over the 10-step telemetry interval is 0.14% of training wall-clock. Deep-tail evaluation is the second-largest line item at roughly 4 GPU-hours per checkpoint; it is the reason the dose grid is as coarse as it is.