UNDERSTANDING DIVERSITY COLLAPSE IN RLVR VIA THE LENS OF OVERTRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) often suffers from diversity collapse: Pass@1 improves while high-k Pass@k degrades. We study this divergence through the lens of overtraining. We first show that validation Pass@1 largely plateaus while training-set success continues to climb and high-k coverage continues to decline. We then show that a problem can retain substantial Pass@1 headroom after its direct contribution to high-k coverage has nearly saturated. A Bayesian analysis quantifies this distinction under few-rollout uncertainty. Early stopping reduces coverage loss. We also find that stopping updates on high-success problems preserves or improves Pass@256 relative to the base model while retaining Pass@1 gains, even as training continues on the remaining problems. Together, these results identify overtraining as a source of diversity collapse in RLVR: continued optimization outlasts its coverage benefit, and restricting its period or scope reduces the resulting coverage loss.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has emerged as a particularly effective paradigm for improving the reasoning ability of large language models (LLMs) on tasks with objectively checkable outcomes (Shao et al., 2024; Lambert et al., 2025; Guo et al., 2025). Despite these advances, RLVR is also known to suffer from *diversity collapse*: while Pass@1 improves, high-k Pass@k frequently degrades as training progresses (Yue et al., 2025; Zhu et al., 2025; Li et al., 2026). Since high-k Pass@k measures whether the model can find a correct solution within k attempts, this degradation is often viewed as a narrowing of the model's reasoning boundary (Tuyls et al., 2025; Yue et al., 2025). One line of explanation attributes diversity collapse to the self-reinforcing nature of on-policy RLVR updates (Mahdavi et al., 2025; Walder and Karkhanis, 2025; Chen et al., 2025b; Thrampoulidis et al., 2025): trajectories that are already more likely to succeed are sampled and reinforced more often, creating a sharpening bias that concentrates probability mass on a shrinking subset (Jin et al., 2025; Fan et al., 2026; Li et al., 2026). This sharpening bias, however, does not by itself explain when an update still improves coverage and when it merely reshapes probability mass among already-covered problems.

In this paper, we study this distinction through the lens of *overtraining*. We first observe that current training recipes overtrain for Pass@1 itself: validation Pass@1 largely plateaus after the first few epochs, even as training-set success continues to climb throughout the remaining schedule. The later stages of standard RLVR training contribute little to Pass@1 while largely degrading high-k Pass@k (Figure 2). On the MinervaMath and OlympiadBench average, training from epoch 5 to epoch 20 improves Pass@1 only from 37.3% to 37.7%, while Pass@256 drops from 78.4% to 75.8%.

If we shift the reference objective from Pass@1 to Pass@k for large k, the scope of overtraining turns out to be far broader. For high-k Pass@k, saturation is aggressive (Figure 1): since Pass@k equals $1-(1-p)^k$ for a problem with success probability p, it saturates rapidly once p becomes non-negligible. A problem can therefore remain far from Pass@1 saturation while its direct contribution to high-k coverage is already nearly saturated. Further updates can still improve its single-attempt success probability while adding little coverage. We quantify this remaining coverage benefit from the success counts observed in a few training rollouts, using a standard Beta–Binomial model. Under a uniform prior with $n = 8$ rollouts, the posterior expected marginal utility for Pass@256 with one observed success is only about $1/33$ of that with no observed successes.

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.

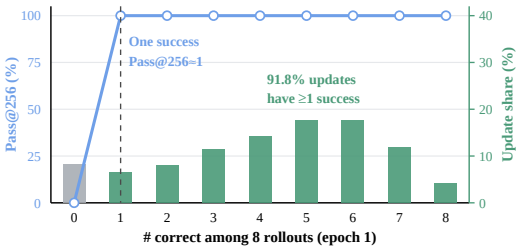


Figure 1: *High-k coverage saturates at low success rates.* The blue curve evaluates Pass@256 at the plug-in estimate $\hat{p} = m/8$, where m counts successes among eight rollouts. The bars show epoch-1 prompt shares: 91.8% have at least one observed success.

Observationally, we track problems with no observed success under the base policy (Pass@256 $= 0$) and find that a non-trivial fraction produce correct responses early in training, even as a larger number of previously covered problems lose their observed coverage. Aggregate coverage decline thus contains both newly observed successes and losses of previously observed successes. The epoch-level overtraining identified above motivates *early stopping*, which recovers much of the high-k Pass@k lost in later training while retaining most of the Pass@1 gain.

Interventionally, restricting updates to problems with zero observed success lifts Pass@256 above the base model on difficult benchmarks, while higher-success buckets generally retain less coverage and produce stronger entropy contraction. These results suggest that early stopping need not terminate the entire training run: updates can stop on high-success problems while continuing on the rest.

We test this form of problem-level early stopping using the posterior utility to select which problems to keep updating. In this experiment, termed *Bayesian Boundary Gating* (BBG), problems below a utility threshold are excluded from updates entirely, while the remaining problems receive equal problem-level weights. Across multiple reasoning benchmarks, training can continue on this retained set while preserving or improving Pass@256 relative to the base model and retaining Pass@1 gains.

Our main contributions can be summarized as follows:

1. We study diversity collapse in RLVR through the lens of *overtraining*, first identifying continued coverage loss after validation Pass@1 largely plateaus, then showing why a problem's direct coverage benefit can saturate well before its single-attempt accuracy.
2. We provide both observational and interventional evidence locating coverage loss across training stages and update scopes: newly observed successes coexist with losses of previously observed successes, and restricting updates to different rollout-success buckets produces different coverage and entropy outcomes.
3. We show that early stopping reduces coverage loss both by ending training sooner and by stopping updates on high-success problems while training continues on the rest.

2 Related Work

RLVR. Open systems such as Tulu 3, DeepSeekMath, DeepSeek-R1, and DAPO established the modern recipe space for RLVR (Lambert et al., 2025; Shao et al., 2024; Guo et al., 2025; Yu et al., 2025). A complementary line of work studies how to allocate limited training budget more effectively, including curriculum learning, selective rollouts, online difficulty filtering, and experience reuse (Chen et al., 2025a; Zheng et al., 2025; Wang et al., 2025; Bae et al., 2026; Zhan et al., 2025). These works primarily target sample and training efficiency; by contrast, we take the evolving learning state of each problem and its consequence for high-k Pass@k as the central object of study.

Diversity Collapse and the Reasoning Boundary. Reinforcement learning from human feedback has long been observed to reduce output diversity (Kirk et al., 2024; Xiao et al., 2024). Within RLVR, this issue is especially consequential because reduced diversity can lower multi-attempt success. Recent work characterizes the phenomenon as reasoning boundary shrinkage or capability boundary collapse (Wen et al., 2025; Nguyen et al., 2025; Dong et al., 2025; Fan et al., 2026), and Yue et al. (2025) uses the aggregate decline of high-k Pass@k to argue that RLVR does not expand reasoning capacity beyond the base model. Closely related, Le et al. (2025) and Liu et al. (2025) revisit prompts with weak or zero within-group reward variance, showing that such prompts should not be uniformly discarded. We study diversity collapse from an overtraining perspective, linking it to both epoch-level and problem-level overtraining. We find that newly observed successes coexist with losses of previously covered problems, even when aggregate Pass@k declines.

Pass@k-aware Objectives. Several recent methods revisit reward design, advantage estimation, or gradient reweighting to better preserve solution diversity and improve Pass@k (Tang et al., 2025; Mahdavi et al., 2025; Walder and Karkhanis, 2025; Chen et al., 2025b; Thrampoulidis et al., 2025; Peng et al., 2026). Yu (2025) argues that Pass@k is better viewed as a diagnostic of exploration than as a training objective. MaxRL (Tajwar et al., 2026) approximates maximum likelihood with a finite-rollout objective, giving harder problems greater weight. It changes the optimization objective; we study when problems with little remaining coverage benefit should leave the optimization scope. We find that the direct coverage benefit can saturate while substantial Pass@1 headroom remains.

Over-optimization and Entropy Collapse. Mismatches between proxy and true objectives can induce Goodhart-style degradation and reward hacking (Amodei et al., 2016; Skalse et al., 2022; Gao et al., 2023; Karwowski et al., 2024). In the RLVR setting, this concern manifests in entropy collapse, sampling bias, and divergence choice (Jin et al., 2025; Fan et al., 2026; Li et al., 2026). Existing mitigations regulate entropy or apply mass-covering regularization and alternative f-divergences to the high-success subset (Jin et al., 2025; Nguyen et al., 2025; Li et al., 2026); relatedly, emphasizing negative reinforcement can improve Pass@k across a broad range of k (Zhu et al., 2025; Yao et al., 2026). These approaches address diversity collapse through objective design or regularization. We show that continued optimization can have little remaining coverage benefit, and that early stopping and removing high-success problems from updates reduce coverage loss.

2.1 A Simplified View of RLVR

Let $\mathcal{D}$ denote a distribution over training problems x, where each problem is instantiated as a prompt to the language model, and let π_θ be a language model policy over full response trajectories $y = (y_1, \ldots, y_T)$, where $\pi_\theta(y \mid x)$ denotes the probability of generating y given prompt x. For tasks with verifiable outcomes, a deterministic verifier assigns a binary reward $r(x, y) \in \{-1, +1\}$, indicating whether the response is accepted by the answer verifier. While practical RLVR implementations may additionally employ clipped surrogates, KL regularization (Schulman et al., 2017), or group-relative reward normalization (Shao et al., 2024), the common signal shared across these methods is still induced by binary correctness under the current policy. We therefore isolate the verifier-induced success variable $s(x, y) := \frac{r(x,y)+1}{2} \in \{0, 1\}$, and define the success probability of problem x under the current policy as

$$p_\theta(x) := \mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[s(x, y)]. \tag{1}$$

Equivalently, if $\mathcal{C}(x) := \{y : s(x, y) = 1\}$ denotes the set of correct trajectories for problem x, then

$$p_\theta(x) = \sum_{y\in\mathcal{C}(x)} \pi_\theta(y \mid x). \tag{2}$$

The verifier-induced component of RLVR can thus be written as

$$J_{\text{ver}}(\theta) := \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}[s(x, y)] = \mathbb{E}_{x\sim\mathcal{D}}[p_\theta(x)]. \tag{3}$$

This is the average probability that the current policy solves a randomly drawn training problem. Denoting by Pass@k the probability that at least one of k independent samples is correct, $J_{\text{ver}}(\theta)$ coincides with Pass@1.

Rollout Success. During training, RLVR does not observe $p_\theta(x)$ directly. Instead, for each problem it only sees a finite number of on-policy rollouts. We therefore define the empirical success rate

$$\hat{p}_{\theta,n}(x) := \frac{1}{n}\sum_{i=1}^{n} s(x, y_i), \qquad y_i \sim \pi_\theta(\cdot \mid x), \tag{4}$$

where n is the number of sampled trajectories for problem x in training. This statistic is quantized at resolution $1/n$. For example, when $n = 8$, training can only distinguish the discrete success buckets $\{0/8, 1/8, \ldots, 8/8\}$. For $m \in \{0, \ldots, n\}$, we define the corresponding rollout-success bucket

$$B_{m/n}(\theta) := \left\{x : \hat{p}_{\theta,n}(x) = \frac{m}{n}\right\}. \tag{5}$$

Since rollout success depends on the current policy, bucket membership changes as θ evolves during training, and we abbreviate $B_{m/n}(\theta)$ to $B_{m/n}$, e.g. $B_{0/8}, B_{1/8}, \ldots, B_{8/8}$ for $n = 8$.

3 OVERTRAINING IN RLVR

3.1 OVERTRAINING FOR PASS@1

Following the practice of Zhu et al. (2025), we train Qwen2.5-Math-7B (Yang et al., 2024) on the MATH training set (Hendrycks et al., 2021) (≈7,500 problems) for 20 epochs with standard RLVR (Eq. (3), $n = 8$ rollouts per problem). We use the verl framework (Sheng et al., 2025), a peak learning rate of 1×10^{-6} with cosine decay, a training batch size of 1024, and a mini-batch size of 128. The KL coefficient is set to 0. The full schedule comprises 140 steps on four H100 GPUs; method-specific configurations are given in Appendix A.

For each evaluation benchmark, we generate 256 independent samples per problem using temperature $T = 0.6$, nucleus sampling with top-$p = 0.95$, and a maximum generation length of 4096 tokens. Pass@k is computed using the unbiased estimator of Chen et al. (2021). Evaluation uses the *any-boxed* extraction rule: a response is counted as correct if any boxed expression matches the ground truth after normalization. At $k = 256$, empirical coverage is the fraction of problems with at least one accepted response. All coverage comparisons below use this verifier and sampling protocol.

We track training-set and validation performance during training. In early training (epochs 1–5), the training-set rollout success rate and validation Pass@1 (MATH-500; Lightman et al. (2024)) both improve together. After epoch 5, however, a characteristic overfitting pattern emerges (Figure 2(a)): training-set success continues to climb, while validation Pass@1 largely plateaus. This divergence suggests that the policy learns to solve the same training problems more reliably, but these additional training-set gains yield little improvement in validation Pass@1. At the same time, high-k coverage continues to decline. As shown in Figure 2(b), on the MinervaMath+OlympiadBench average, training from epoch 5 to epoch 20 improves Pass@1 only from 37.3% to 37.7%, while Pass@256 drops from 75.2% to 70.6%. The later coverage loss is therefore accompanied by little additional validation accuracy gain.



(a) Training and validation diverge

(b) Late training continues to lose coverage

Figure 2: *Overtraining in the later stages of RLVR.* (a) Training-set success continues to increase after validation Pass@1 on MATH-500 largely plateaus. (b) On the MinervaMath and OlympiadBench average, epoch 5 to epoch 20 provides only 0.4 percentage points of additional Pass@1 while losing 4.6 points of Pass@256. Both panels use Qwen2.5-Math-7B trained on MATH.

Early in training, a loss of coverage may accompany a substantial improvement in Pass@1. The later stage differs because coverage continues to decline after the additional accuracy gain has become small. Thus, even if one accepts an initial accuracy–coverage trade-off, the remaining schedule provides much less additional accuracy for the coverage it continues to lose.

These observations identify overtraining at the *epoch level*. As training progresses, the combined share of the $B_{7/8}$ and $B_{8/8}$ buckets rises from 16.0% to 64.3%, showing that sampled training problems increasingly occupy near-resolved states.

3.2 OVERTRAINING FOR PASS@k

The analysis above takes $J_{\text{ver}} = J_1$ (Pass@1) as the reference and identifies overtraining in the late training epochs. We now show that, if we regard high-k Pass@k as the reference metric, the scope of low-return optimization extends far beyond the near-resolved regime. A problem can retain substantial Pass@1 headroom after its direct contribution to high-k coverage has nearly saturated.

Define the Pass@k objective over the training distribution as

$$J_k(\theta) \;:=\; \mathbb{E}_{x\sim\mathcal{D}}\Big[1-\big(1-p_\theta(x)\big)^k\Big]. \tag{6}$$

This is the expected probability that at least one of k independent samples solves problem x. The marginal value of a unit increase in $p_\theta(x)$ under this objective is

$$\frac{\partial}{\partial p_\theta(x)}\Big[1-\big(1-p_\theta(x)\big)^k\Big] = k\big(1-p_\theta(x)\big)^{k-1}, \tag{7}$$

which decays exponentially once $p_\theta(x)$ exceeds a small threshold. For Pass@1, the corresponding derivative is always 1. Thus, the same increase in a problem's success probability can still improve single-attempt accuracy while adding little high-k coverage (Figure 6 in Appendix B). Relative to coverage, continued optimization can enter this low-return regime before the problem reaches $7/8$ or $8/8$ observed successes.

For example, a problem with true success probability $p = 0.05$ has Pass@1 of only 0.05, but Pass@256 is already $1 - 0.95^{256} \approx 0.999998$. Its remaining single-attempt headroom is 0.95, while its remaining coverage headroom is approximately 2×10^{-6}. Further increases in the same success probability can still improve Pass@1 while contributing almost nothing to this problem's Pass@256.

In practice, RLVR typically uses a small number of rollouts per problem, $n = 8$ (Shao et al., 2024) or $n = 16$ (Guo et al., 2025), so the training procedure observes success only at a coarse resolution of $1/n$. The empirical rate $\hat{p}_{\theta,n}(x) = m/n$ is not the true success probability $p_\theta(x)$. We therefore adopt a standard Beta–Binomial Bayesian model: placing a uniform prior on p and conditioning on m successes out of n rollouts yields $p \mid m, n \sim \mathrm{Beta}(m+1, n-m+1)$. The expected marginal utility under this posterior, which we call the *Bayesian boundary utility*, is

$$u(m;n,k) := \mathbb{E}_{p\sim\mathrm{Beta}(m+1,n-m+1)}\big[k(1-p)^{k-1}\big] = k\prod_{j=1}^{k-1}\frac{n-m+j}{n+1+j}, \tag{8}$$

where the closed form follows from the moments of the Beta distribution (Gelman et al., 1995). Intuitively, $u(m;n,k)$ measures the expected marginal improvement in Pass@k from a unit increase in $p_\theta(x)$, averaged over posterior uncertainty about p.

Under this posterior, the ratio between the one-success and zero-success buckets simplifies to

$$\frac{u(1;n,k)}{u(0;n,k)} = \frac{n}{n+k-1}. \tag{9}$$

For $n = 8$ and $k = 256$, $u(0) \approx 8.73$, $u(1) \approx 0.265$, and $u(2) \approx 0.0071$. The posterior expected marginal utility for Pass@256 with one success among eight rollouts is therefore only about $1/33$ *of the zero-success value* (Figure 3). For Pass@1, by contrast, $u(m;n,1) = 1$ for every bucket. The same rollout evidence thus implies very different remaining direct benefits under the two metrics.

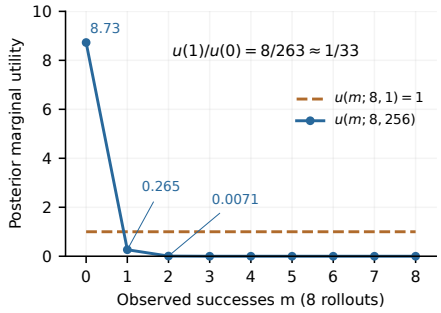


Figure 3: *Posterior marginal utility.* With a uniform prior and $n = 8$ rollouts, the curves show the expected marginal gain in Pass@k from increasing a problem's true success probability. For Pass@256, utility falls from 8.73 at $m = 0$ to 0.265 at $m = 1$, about 33-fold, and to 0.0071 at $m = 2$. Pass@1 utility remains 1 across all buckets, revealing different remaining direct benefits under the same rollout evidence.

At epoch 1, 91.8% of sampled training problems already have at least one observed success (Figure 1), so most sampled problems have much lower posterior marginal utility for Pass@256 than $B_{0/8}$. Their contribution to the gradient also depends on the advantage rule. Continuing to optimize these problems can still improve Pass@1, but their remaining direct coverage benefit is small. The zero-success bucket retains the largest posterior utility, although it remains a noisy proxy for problems with low coverage when $n \ll k$ (Appendix B).

The two forms of overtraining therefore concern different aspects of the same process:

1. **Training continues beyond validation gains.** Later epochs provide little additional Pass@1 improvement while high-k coverage continues to decline.
2. **Coverage gains can be exhausted much earlier.** At the problem level, increasing success probability can retain substantial Pass@1 after its direct high-k contribution is nearly saturated.

The actual change in p depends on the update and its effects through shared parameters. Along a differentiable training trajectory θ_t, with the problem distribution held fixed,

$$\frac{dJ_k(\theta_t)}{dt} = \mathbb{E}_{x\sim\mathcal{D}}\left[k\big(1-p_{\theta_t}(x)\big)^{k-1}\frac{dp_{\theta_t}(x)}{dt}\right]. \tag{10}$$

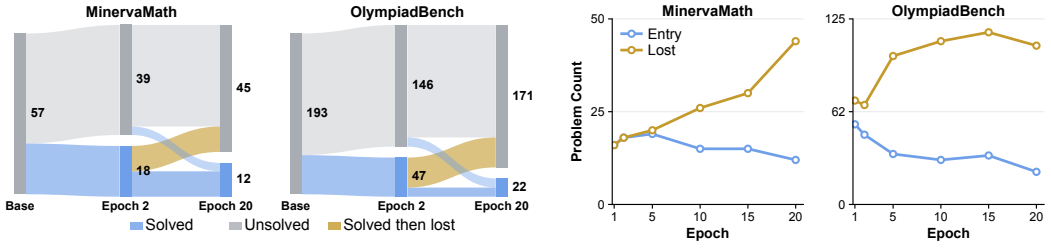
Gains on high-success problems have little weight in this derivative, while losses on low-success problems can have much greater weight. If continued optimization also reduces success probabilities on low-coverage problems, these losses can outweigh the direct gains on already-covered problems.

4 Locating and Reducing Overtraining

To test the overtraining explanation for coverage loss, we track coverage gains and losses during training and examine the effects of early stopping and excluding high-success problems from updates. We compare three update rules: (i) *REINFORCE* (Williams, 1992), which applies the standard policy gradient with binary reward $r \in \{-1, +1\}$; (ii) *GRPO* (Shao et al., 2024), which normalizes advantages within each problem's rollout group; and (iii) *W-RF* (Zhu et al., 2025), which emphasizes negative reinforcement by reweighting the advantage to mitigate diversity collapse. We use REINFORCE for coverage tracking, early stopping, bucket-restricted training, and progressive masking.

4.1 Coverage Dynamics and Early Stopping

To distinguish coverage gains from losses during training, we track individual problems relative to the base policy. For each benchmark, we identify the subset of problems with no observed success under the base policy (Pass@256 $= 0$) and re-evaluate this fixed subset at each training epoch. We also track problems with an observed base-model success that later produce no correct responses. The number of entries minus the number of losses, divided by the evaluation-set size, gives the change in empirical Pass@256.



(a) Coverage across epochs

(b) Entry and loss across epochs

Figure 4: *Newly observed coverage is increasingly offset by coverage loss.* (a) Sankey diagram tracking problems by observed Pass@256 status across epochs: some problems with no base-model success become covered, but part of this coverage is lost during later training. (b) Entry and loss counts at each checkpoint relative to the base evaluation. "Solved" in (a) denotes at least one observed correct response in 256 samples. See Appendix C for a sampling-noise analysis and Appendix D for additional results.

As shown in Figure 4, a non-trivial fraction of problems with no observed base-model success produce correct responses early in training. This gain is not visible in aggregate Pass@256 because it is offset by a larger number of previously covered problems losing coverage as training progresses.

The counts make this offset explicit. Across the held-out problems analyzed in Appendix C, 70 of 265 base zero-success problems have an observed success at epoch 2; this number falls to 38 at epoch 20.

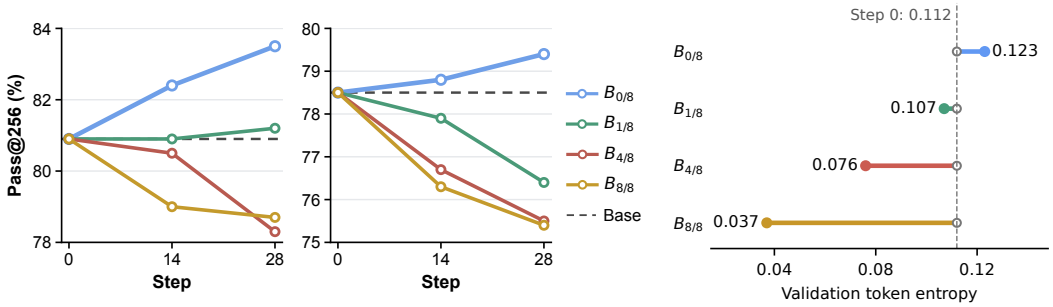
At the final checkpoint, these 38 entries coexist with 152 losses among the 711 problems covered by the base evaluation. The changes also include multi-success problems: 27 of the early entries produce at least two correct samples, and 91 of the final losses had at least two correct samples under the base model. The appendix gives the success-count and persistence breakdowns.

Early stopping. Together with the validation plateau in Section 3.1, this accumulation of coverage loss motivates truncating the later stages of training. We select the epoch 5 and epoch 10 checkpoints from the same 20-epoch REINFORCE run (Table 1). Early stopping at 5 epochs already recovers much of the high-k Pass@k lost in later training, e.g. raising MinervaMath Pass@256 from 77.9 under full REINFORCE back to 79.4, close to the base model's 80.9, while retaining most of the Pass@1 gain. Pass@1 changes from 38.9 to 36.0, still well above the base model's 24.4. On OlympiadBench, stopping at epoch 5 similarly raises Pass@256 from 73.7 to 77.4, while Pass@1 changes from 38.5 to 36.3. Thus, reducing training period preserves coverage after additional validation accuracy gains have become small.

4.2 Bucket-Restricted Training and Masking

The early-stopping results show that coverage loss can be reduced by shortening training. To examine whether coverage can also be preserved while training continues, we vary which rollout-success buckets receive updates.

Bucket-restricted RLVR. To distinguish the consequences of different update scopes, we train variants in which updates are restricted to a single rollout-success bucket $B_{m/8}$ for $m \in \{0, 1, 4, 8\}$, and compare Pass@256 to the base model. These runs use REINFORCE with signed rewards: for problems in $B_{0/8}$, all rollouts receive a negative-reinforcement signal. Standard GRPO's group-relative centering would instead assign zero advantage to this bucket. By the analysis of Section 3.2, $B_{0/8}$ carries the largest posterior marginal utility for high-k coverage. As shown in Figure 5, restricting updates to $B_{0/8}$ improves Pass@256 over the base model on both benchmarks, while higher-success buckets generally retain less coverage.



(a) MinervaMath (left) and OlympiadBench (right)

(b) Entropy at step 28

Figure 5: *Coverage and entropy respond differently to the update scope.* (a) Pass@256 improves when updates are restricted to $B_{0/8}$, while higher-success buckets generally retain less coverage. (b) Validation token entropy at step 28, compared with the common step-0 value 0.112. Higher-success buckets produce progressively stronger entropy contraction.

Entropy evidence. The same bucket interventions let us examine whether the coverage contrast is accompanied by policy sharpening. From the common step-0 value 0.112, validation token entropy at step 28 becomes 0.123, 0.107, 0.076, and 0.037 for $B_{0/8}$, $B_{1/8}$, $B_{4/8}$, and $B_{8/8}$ respectively. Higher-success interventions therefore exhibit stronger entropy contraction alongside generally worse coverage retention, consistent with continued policy sharpening. The coverage ordering is not strictly monotone: $B_{1/8}$ remains slightly above the base model on MinervaMath.

Progressive bucket masking. We therefore progressively exclude $B_{8/8}$, $B_{\geq 7/8}$, and $B_{\geq 6/8}$ from full training, retaining equal problem-level coefficients on the remaining prompts (Table 1). Masking $B_{\geq 6/8}$ raises MinervaMath Pass@256 from 77.9 to 80.8, while Pass@1 changes from 38.9 to 35.0. On OlympiadBench, it raises Pass@256 from 73.7 to 77.2 while preserving nearly all of the full-training Pass@1 (38.5 versus 37.9).

Table 1: Effect of early stopping and progressive bucket masking. Each cell reports Pass@1 / Pass@256 / Avg., where Avg. is the mean over the nine evaluated budgets. All results use Qwen2.5-Math-7B. Retained prompts have equal problem-level coefficients. Bold marks the best value for each metric within each benchmark.

Intervention	MATH-500	OlympiadBench	MinervaMath	AIME 2025
Base Model	64.5 / 96.2 / 86.8	28.5 / **78.5** / 58.0	24.4 / **80.9** / 57.6	7.4 / 53.3 / 27.5
REINFORCE (Williams, 1992)	75.5 / 92.0 / 85.9	38.2 / 68.7 / 55.8	37.3 / 72.4 / 57.9	10.4 / 50.0 / 26.6
GRPO (Shao et al., 2024)	**77.5** / 93.0 / 87.2	**39.6** / 71.5 / 58.3	**39.2** / 74.6 / 58.7	**10.8** / 46.7 / 27.7
W-RF (Zhu et al., 2025)	76.5 / 94.4 / 87.9	38.5 / 73.7 / 59.4	38.9 / 77.9 / 61.2	9.7 / 53.3 / 31.3
+ early stop (10 ep)	75.6 / 96.0 / **88.9**	37.3 / 74.9 / 59.9	37.6 / 78.3 / 61.3	10.2 / 56.7 / **32.3**
+ early stop (5 ep)	74.0 / **96.6** / **88.9**	36.3 / 77.4 / **60.7**	36.0 / 79.4 / 61.6	9.1 / 53.3 / 29.6
+ mask $B_{8/8}$	76.2 / 94.4 / 88.4	36.5 / 77.6 / 60.5	32.6 / 80.2 / 61.4	9.2 / **60.0** / 31.5
+ mask $B_{\geq 7/8}$	74.5 / 95.6 / 88.7	37.6 / 76.4 / 60.5	34.8 / 80.6 / **61.7**	9.8 / **60.0** / 31.3
+ mask $B_{\geq 6/8}$	73.6 / 96.0 / 88.8	37.9 / 77.2 / 60.7	35.0 / 80.8 / **61.7**	9.7 / 53.3 / 31.1

4.3 Early Stopping at the Problem Level

To select which problems remain active during training, we use their posterior marginal utility for high-k coverage. This is a form of early stopping at the problem level: updates stop on problems with little remaining coverage benefit while continuing on the rest.

Bayesian boundary utility. To assess how much an update on a given problem can still contribute to high-k coverage, we use the posterior estimate in Eq. (8). Under a uniform prior on p, observing m successes in n rollouts gives $p \mid m, n \sim \mathrm{Beta}(m+1, n-m+1)$. The Bayesian boundary utility $u(m; n, k)$ averages the marginal coverage gain $k(1-p)^{k-1}$ under this posterior.

Bayesian Boundary Gating (BBG). We use $u(m; n, k)$ to identify problems with little remaining coverage benefit and suppress updates on them. Retained problems receive equal problem-level weights, so the intervention tests whether selecting which problems to keep updating preserves coverage. Since $\frac{1}{n+1}\sum_{m=0}^{n} u(m; n, k) = 1$ under the uniform prior, the utility is naturally normalized. For the observed success count m, we set

$$\alpha(m) = \begin{cases} 1, & \text{if } u(m; n, k_{\mathrm{ref}}) \geq \tau(n+1), \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Buckets whose share of the total utility falls below τ are hard-gated and excluded from backpropagation entirely. Because α is fixed given (m, n, k_{ref}) and τ, the cutoff does not depend on the bucket distribution within a particular batch. The stopping decision is recomputed from each problem's current rollout-success bucket; all retained problems have $\alpha = 1$.

For the controlled intervention, we use $n = 8$, $k_{\mathrm{ref}} = 16$, and $\tau = 0.01$. This retains $B_{0/8}$ through $B_{3/8}$ and excludes $B_{\geq 4/8}$, or equivalently $\alpha(m) = \mathbf{1}[m \leq 3]$. The reference $k_{\mathrm{ref}} = 16$ sits at the geometric midpoint of the evaluated range $k \in [1, 256]$; the $k = 256$ example in Section 3.2 illustrates the sharper utility decay at a larger reference budget. We apply the gate to W-RF without changing its trajectory-level loss (Appendix A.4).

Coverage retention across training sets. In the controlled 7B comparison, stopping updates according to posterior utility preserves or improves Pass@256 relative to the base model on all four benchmarks, while retaining Pass@1 gains. Table 2 extends the comparison to Qwen2.5-Math-1.5B (Yang et al., 2024) trained on MATH (Hendrycks et al., 2021) and POLARIS-53K (An et al., 2025). It includes REINFORCE, GRPO, and MaxRL in both settings. The BBG runs retain coverage above the base model in all eight training-set/benchmark pairs, together with Pass@1 gains, although the standard objectives often reach higher Pass@1.

Answer extraction and verifier behavior. Our Pass@k evaluation uses an *any-boxed* extraction rule: a response is counted as correct if any boxed expression matches the ground truth after normalization. This relaxes only the answer-placement convention, not the correctness criterion. Empirically, this change improves the *base model* across all benchmarks and also benefits trained models, especially on *MinervaMath*, suggesting that some apparent errors are due to answer exposure and formatting rather than missing mathematical content. This also provides a useful lens on recent findings that RLVR can improve performance with extremely small or imperfect training signals (Wang et al.,

Table 2: Coverage retention across two training sets. Qwen2.5-Math-1.5B results; each cell reports Pass@1 / Pass@256 / Avg. in percent. Avg. is the arithmetic mean over $k \in \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$. Bold marks the best Pass@256 in each setting.

Update rule	MATH-500	OlympiadBench	MinervaMath	AIME 2025
MATH (step 140)				
Base Model (Yang et al., 2024)	42.9 / 95.4 / 79.8	22.6 / 73.8 / 51.8	10.9 / 72.4 / 44.2	4.9 / 50.0 / 24.0
REINFORCE (Williams, 1992)	69.3 / 93.6 / 84.8	31.8 / 68.9 / 52.6	30.6 / 72.1 / 53.2	6.8 / 50.0 / 28.7
GRPO (Shao et al., 2024)	71.1 / 94.6 / 86.3	32.7 / 73.0 / 55.0	31.7 / 71.3 / 54.5	7.0 / 50.0 / 27.2
MaxRL (Tajwar et al., 2026)	70.4 / 94.8 / 86.4	32.4 / 72.4 / 55.1	31.4 / 74.3 / 54.8	6.7 / 50.0 / 27.5
BBG (§4.3)	64.4 / **96.0** / 85.3	28.8 / **74.1** / 54.3	26.0 / **77.6** / 54.6	5.6 / **56.7** / 27.3
POLARIS-53K (500 rollout/forward steps)				
Base Model (Yang et al., 2024)	42.9 / 95.2 / 79.8	22.6 / 73.8 / 51.8	10.9 / 72.4 / 44.2	4.9 / 50.0 / 24.0
REINFORCE (Williams, 1992)	67.8 / 95.0 / 85.7	31.0 / 72.3 / 54.2	30.7 / 74.6 / 55.1	5.5 / 46.7 / 25.3
GRPO (Shao et al., 2024)	68.4 / 95.2 / 85.9	31.3 / 73.0 / 54.7	31.3 / 75.4 / 55.2	5.1 / 43.3 / 22.5
MaxRL (Tajwar et al., 2026)	68.1 / 95.4 / 86.2	31.6 / 73.0 / 55.2	31.2 / 76.1 / 56.0	6.2 / 46.7 / 25.7
BBG (§4.3)	63.4 / **95.8** / 85.0	28.3 / **74.7** / 54.3	24.5 / **79.8** / 54.7	5.3 / **60.0** / 27.4

2025; Shao et al., 2026): a boxed-answer verifier rewards not only reasoning, but also producing answers in the form expected by the evaluator. Conversely, methods that prioritize output diversity may be disadvantaged by such format-sensitive protocols, because they optimize less directly for a single canonical answer presentation. We therefore interpret short-RLVR gains partly as improved instruction following and answer exposure, distinct from genuine reasoning-boundary expansion.

MATH-500 (Lightman et al., 2024). All methods improve Pass@1. The base model's Pass@256 of 95.4% and 95.2% in the two settings leaves little room for further coverage improvement, but it does not prevent coverage loss: on MATH training, REINFORCE, GRPO, and MaxRL fall to 93.6, 94.6, and 94.8, respectively. BBG reaches 96.0 and 95.8 under MATH and POLARIS-53K training, while retaining Pass@1 gains from 42.9 to 64.4 and 63.4. These gains are smaller than those of the strongest Pass@1 baselines, making the accuracy–coverage trade-off explicit.

OlympiadBench & MinervaMath (He et al., 2024; Lewkowycz et al., 2022). Diversity collapse emerges clearly on OlympiadBench: every non-BBG training objective improves Pass@1 but ends below the base model at Pass@256 in both training settings. BBG instead reaches 74.1 and 74.7, above the base value of 73.8. On MinervaMath, BBG reaches 77.6 and 79.8, exceeding both the base model's 72.4 and the strongest alternative, MaxRL, at 74.3 and 76.1. This coverage retention comes with lower Pass@1 than the strongest accuracy baseline: for example, after training on MATH, BBG attains 26.0 on MinervaMath versus GRPO's 31.7.

AIME 2025 (Zhu et al., 2025). The base model has Pass@256 of 50.0 in both settings. BBG raises this to 56.7 with MATH training and 60.0 with POLARIS-53K training; the other methods match or fall below the base at $k = 256$. AIME's low single-sample success rates make the sampling budget especially relevant: as k increases, Pass@k saturates at progressively smaller $p_\theta(x)$, extending the low-marginal-utility regime to harder problems.

5 Conclusion

We studied diversity collapse in RLVR through the lens of overtraining. We first showed that later training provides little additional validation Pass@1 improvement while high-k coverage continues to decline. We then showed that the scope of overtraining is broader from the coverage perspective: a problem can retain substantial single-attempt accuracy headroom after its direct contribution to high-k coverage has nearly saturated. Posterior marginal utility makes this distinction explicit under few-rollout uncertainty. Coverage tracking and bucket-restricted training connect the two levels to the observed losses, while early stopping and hard masking reduce those losses. Early stopping can therefore be applied to individual problems while training continues on the rest. Together, these results identify overtraining as a source of diversity collapse in RLVR: continued optimization outlasts its coverage benefit, and restricting its period or scope reduces the resulting coverage loss.

AI USE STATEMENT

Language models and AI agents were used for coding assistance, grammar polishing, and phrasing refinement during the drafting of this paper. While artificial intelligence assisted in the presentation, all remaining errors, oversights, and bugs remain the exclusive product of natural stupidity.

REPRODUCIBILITY STATEMENT

Experiments use Qwen2.5-Math-7B and 1.5B (Yang et al., 2024) on MATH (Hendrycks et al., 2021), with additional 1.5B experiments on POLARIS-53K (An et al., 2025). The controlled 7B experiments use the verl framework (Sheng et al., 2025) and vLLM (Kwon et al., 2023), with four NVIDIA H100 GPUs for training and one for evaluation. We evaluate on MATH-500, OlympiadBench, MinervaMath, and AIME 2025. The main text reports training budgets, rollout counts, early-stopping checkpoints, and bucket-selection rules. Appendix A provides the training hyperparameters, method-specific configurations, and evaluation protocol.

REFERENCES

Alan Agresti. *Categorical data analysis*. John Wiley & Sons, 2013.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Chenxin An, Zhihui Xie, Xiaonan Li, Lei Li, Jun Zhang, Shansan Gong, Ming Zhong, Jingjing Xu, Xipeng Qiu, Mingxuan Wang, and Lingpeng Kong. POLARIS: A post-training recipe for scaling reinforcement learning on advanced reasoning models, 2025. URL https://hkunlp.github.io/blog/2025/Polaris.

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning oriented reinforcement learning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 700–719, 2026.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021.

Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. Self-evolving curriculum for llm reasoning. *arXiv preprint arXiv:2505.14970*, 2025a.

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*, 2025b.

Yihong Dong, Xue Jiang, Yongding Tao, Huanyu Liu, Kechi Zhang, Lili Mou, Rongyu Cao, Yingwei Ma, Jue Chen, Binhua Li, et al. Rl-plus: Countering capability boundary collapse of llms in reinforcement learning with hybrid-policy optimization. *arXiv preprint arXiv:2508.00222*, 2025.

Mingyuan Fan, Weiguang Han, Daixin Wang, Cen Chen, Zhiqiang Zhang, and Jun Zhou. When sharpening becomes collapse: Sampling bias and semantic coupling in rl with verifiable rewards. *arXiv preprint arXiv:2601.15609*, 2026.

R. A. Fisher. On the interpretation of χ^2 from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 85(1):87–94, 1922. ISSN 09528385. URL http://www.jstor.org/stable/2340521.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

Daya Guo, Dejian Yang, Haowei Zhang, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems, 2024.

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 *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

Renren Jin, Pengzhi Gao, Yuqi Ren, Zhuowen Han, Tongxuan Zhang, Wuwei Huang, Wei Liu, Jian Luan, and Deyi Xiong. Revisiting entropy in reinforcement learning for large reasoning models. *arXiv preprint arXiv:2511.05993*, 2025.

Jacek Karwowski, Oliver Hayman, Xingjian Bai, Klaus Kiendlhofer, Charlie Griffin, and Joar Max Viktor Skalse. Goodhart's law in reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024.

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 *The Twelfth International Conference on Learning Representations*, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph 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*, SOSP '23, page 611–626, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702297. doi: 10.1145/3600006.3613165.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025.

Thanh-Long V Le, Myeongho Jeon, Kim Vu, Viet Lai, and Eunho Yang. No prompt left behind: Exploiting zero-variance prompts in llm reinforcement learning via entropy-guided advantage shaping. *arXiv preprint arXiv:2509.21880*, 2025.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022.

Long Li, Zhijian Zhou, JIARAN HAO, Jason Klein Liu, Yanting Miao, Wei Pang, Xiaoyu Tan, Wei Chu, Zhe Wang, Shirui Pan, Chao Qu, and Yuan Qi. The choice of divergence: A neglected key to mitigating diversity collapse in reinforcement learning with verifiable reward. In *The Fourteenth International Conference on Learning Representations*, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024.

Chenxi Liu, Junjie Liang, Yuqi Jia, Bochuan Cao, Yang Bai, Heng Huang, and Xun Chen. Explore data left behind in reinforcement learning for reasoning language models. *arXiv preprint arXiv:2511.04800*, 2025.

Sadegh Mahdavi, Muchen Li, Kaiwen Liu, Renjie Liao, and Christos Thrampoulidis. Beyond accuracy: A policy gradient reweighting approach for pass@ k maximization in llms. In *2nd AI for Math Workshop@ ICML 2025*, 2025.

Phuc Minh Nguyen, Chinh D La, Duy MH Nguyen, Nitesh V Chawla, Binh T Nguyen, and Khoa D Doan. The reasoning boundary paradox: How reinforcement learning constrains language models. *arXiv preprint arXiv:2510.02230*, 2025.

Ruotian Peng, Yi Ren, Zhouliang Yu, Weiyang Liu, and Yandong Wen. SimKO: Simple pass@k policy optimization, 2026.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. Spurious rewards: Rethinking training signals in rlvr, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, page 1279–1297. Association for Computing Machinery, 2025. ISBN 9798400711961. doi: 10.1145/3689031.3696075.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

Fahim Tajwar, Guanning Zeng, Yueer Zhou, Yuda Song, Daman Arora, Yiding Jiang, Jeff Schneider, Ruslan Salakhutdinov, Haiwen Feng, and Andrea Zanette. Maximum likelihood reinforcement learning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Yunhao Tang, Kunhao Zheng, Gabriel Synnaeve, and Remi Munos. Optimizing language models for inference time objectives using reinforcement learning. In *Forty-second International Conference on Machine Learning*, 2025.

Christos Thrampoulidis, Sadegh Mahdavi, and Wenlong Deng. Advantage shaping as surrogate reward maximization: Unifying pass@ k policy gradients. *arXiv preprint arXiv:2510.23049*, 2025.

Jens Tuyls, Dylan J Foster, Akshay Krishnamurthy, and Jordan T Ash. Representation-based exploration for language models: From test-time to post-training. *arXiv preprint arXiv:2510.11686*, 2025.

Christian Walder and Deep Tejas Karkhanis. Pass@k policy optimization: Solving harder reinforcement learning problems. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*, 2025.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696.

Jiancong Xiao, Ziniu Li, Xingyu Xie, Emily Getzen, Cong Fang, Qi Long, and Weijie J Su. On the algorithmic bias of aligning large language models with rlhf: Preference collapse and matching regularization. *arXiv preprint arXiv:2405.16455*, 2024.

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

Xinhao Yao, Lu Yu, Xiaolin Hu, Fengwei Teng, Qing Cui, JUN ZHOU, and Yong Liu. The debate on RLVR reasoning capability boundary: Shrinkage, expansion, or both? a two-stage dynamic view, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Yang Yu. Pass@ k metric for rlvr: A diagnostic tool of exploration, but not an objective. *arXiv preprint arXiv:2511.16231*, 2025.

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Runzhe Zhan, Yafu Li, Zhi Wang, Xiaoye Qu, Dongrui Liu, Jing Shao, Derek F Wong, and Yu Cheng. Exgrpo: Learning to reason from experience. *arXiv preprint arXiv:2510.02245*, 2025.

Haizhong Zheng, Yang Zhou, Brian R. Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for LLM reasoning via selective rollouts. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

A Training and Evaluation Details

A.1 Base Model and Training Data

The training-dynamics and controlled intervention experiments use Qwen2.5-Math-7B (Yang et al., 2024) as the base policy π_{θ_0}. We train on the MATH training set (Hendrycks et al., 2021), which contains approximately 7,500 competition-level mathematics problems spanning seven subjects (Prealgebra, Algebra, Number Theory, Counting and Probability, Geometry, Intermediate Algebra, and Precalculus). Each problem is paired with a final answer in a canonical form that can be verified by exact string matching after normalization. Each training problem x is formatted as a prompt following the instruction template of the base model. The verifier extracts the final boxed answer from the model's response and compares it against the ground truth using the normalization procedure from Hendrycks et al. (2021).

The comparison in Table 2 uses Qwen2.5-Math-1.5B (Yang et al., 2024) trained on MATH (Hendrycks et al., 2021) and POLARIS-53K (An et al., 2025). Both settings use eight on-policy rollouts per prompt. The MATH runs are compared at step 140; the POLARIS-53K runs use 256 prompts per rollout batch and are compared at 500 rollout/forward steps. BBG buffers active prompts before each backward pass, so the POLARIS budget aligns rollout steps rather than optimizer updates. Evaluation uses the same 256-sample protocol described in Section 3.1. The two groups retain their recorded base evaluations, including MATH-500 Pass@256 of 95.4 and 95.2, respectively.

A.2 Training Infrastructure

The controlled 7B training runs use a single node with 4 NVIDIA H100 96GB GPUs. Evaluation runs use a single H100 GPU. We use the verl framework (Sheng et al., 2025), which decomposes the pipeline into rollout generation and policy optimization. On-policy rollouts are generated using vLLM (Kwon et al., 2023) in bfloat16 precision with eager execution mode and GPU memory utilization set to 0.65; tensor model parallelism is set to 1. The actor model is trained with Fully Sharded Data Parallelism (FSDP) across all 4 GPUs with gradient checkpointing enabled.

A.3 Shared Hyperparameters

Table 3 lists the hyperparameters shared across the controlled 7B methods, following the practice of Zhu et al. (2025). With approximately 7,500 training problems and a batch size of 1024, each epoch corresponds to roughly 7 training steps; the total schedule of 140 steps spans approximately 20 epochs. Following Zhu et al. (2025), we set the KL penalty coefficient to 0, as prior work has found KL regularization can be detrimental in the RLVR setting where the reward signal is exact.

Table 3: Shared training hyperparameters for the controlled 7B experiments. BBG uses an effective batch of 512 active prompts (Appendix A.4).

Hyperparameter	Value
Rollouts per problem (n)	8
Training batch size (prompts)	1024
Mini-batch size (prompts)	128
PPO micro-batch size per GPU	1
Log-prob micro-batch size per GPU	4
Peak learning rate	1×10^{-6}
Learning rate schedule	Cosine decay
KL penalty coefficient	0
Maximum prompt length	1024 tokens
Maximum response length	3072 tokens
Rollout precision	bfloat16
Total training steps	140 ($\approx$ 20 epochs)
Checkpoint / validation frequency	Every 7 steps ($\approx$ 1 epoch)

A.4 METHOD-SPECIFIC CONFIGURATIONS

The configurations below apply to the controlled 7B experiments.

REINFORCE. Uses binary rewards $r(x, y) \in \{-1, +1\}$ with an exponential moving average baseline. No group-level normalization is applied; all $n = 8$ rollouts contribute to the gradient regardless of within-group reward variance.

GRPO. GRPO (Shao et al., 2024) normalizes advantages within each problem's rollout group: $\hat{A}_i = (r_i - \bar{r})/(\sigma_r + \epsilon)$.

W-RF. W-RF (Zhu et al., 2025) scales the advantage for correct rollouts, while incorrect rollouts retain their full advantage weight. This asymmetric weighting partially mitigates diversity collapse by reducing the contribution of positive reinforcement.

BBG. BBG builds on W-RF ($w^+ = 0.3$) and adds problem-level gating based on the Bayesian boundary utility (Eq. (8)). We set the boundary reference k_{ref}=16, the geometric midpoint of the evaluated range $k \in [1, 256]$, and the minimum utility-share threshold τ=0.01. Table 4 lists the per-bucket utility under this configuration: buckets $B_{0/8}$, $B_{1/8}$, $B_{2/8}$, and $B_{3/8}$ are retained with equal problem-level coefficient $\alpha(m) = 1$, while buckets $B_{\geq 4/8}$ fall below the threshold $\tau(n + 1) = 0.09$ and are excluded from backpropagation with $\alpha(m) = 0$. Because gating removes a substantial fraction of the training batch, BBG accumulates active problems into an effective batch of 512 prompts per gradient step. Dropped problems still require rollouts to determine their bucket membership, and BBG may need to sample additional problems before enough active prompts are accumulated. The intervention reallocates optimization effort; the reported results do not measure a wall-clock speedup.

Table 4: Bayesian boundary utility $u(m;\ n$=8, k_{ref}=16) and binary gate for each rollout-success bucket under BBG. Retained problems have equal problem-level weight.

Bucket	$u(m; 8, 16)$	Retained?	$\alpha(m)$
$B_{0/8}$	6.000	Yes	1
$B_{1/8}$	2.087	Yes	1
$B_{2/8}$	0.664	Yes	1
$B_{3/8}$	0.190	Yes	1
$B_{4/8}$	0.047	No	0
$B_{5/8}$	0.010	No	0
$B_{6/8}$	0.0017	No	0
$B_{7/8}$	1.96×10^{-4}	No	0
$B_{8/8}$	1.22×10^{-5}	No	0

A.5 EVALUATION PROTOCOL

For each benchmark, we generate N=256 independent samples per problem with temperature T=0.6, nucleus sampling top-p=0.95, and a maximum generation length of 4096 tokens (longer than the training maximum of 3072 to avoid truncating reasoning chains). Inference uses vLLM in bfloat16 on a single GPU with batch size 1000.

Pass@k is computed using the unbiased estimator of Chen et al. (2021): given N total samples and c correct samples for a problem,

$$\widehat{\text{Pass@}k} \;=\; 1 - \frac{\binom{N-c}{k}}{\binom{N}{k}}, \tag{12}$$

which is unbiased for $1 - (1 - p)^k$ whenever $N \geq k$. We report Pass@k for $k \in \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$; the Avg. column is the arithmetic mean over these nine values. Answer normalization follows Hendrycks et al. (2021); evaluation uses the *any-boxed* extraction rule, counting a response as correct if any boxed expression matches the ground truth. At $k = N = 256$, the estimator is the indicator of at least one observed correct response under this protocol.

A.6 Evaluation Benchmarks

MATH-500 (Lightman et al., 2024) is a 500-problem subset of the MATH test set, serving as the in-distribution benchmark. The base model already achieves Pass@256=96.2%, leaving little room for further coverage improvement. *OlympiadBench* (He et al., 2024) contains 675 olympiad-level problems from international mathematics competitions. *MinervaMath* (Lewkowycz et al., 2022) consists of 272 undergraduate-level quantitative reasoning problems spanning mathematics, physics, and related STEM disciplines. Both are out-of-distribution and contain many problems with low single-sample success but high Pass@256, making them sensitive to diversity collapse. *AIME 2025* (Zhu et al., 2025) consists of the 30 problems from the 2025 American Invitational Mathematics Examination and is the most difficult benchmark. Larger sampling budgets may expose different coverage behavior; Yue et al. (2025) report degradation on AIME at k=1024.

A.7 Further Experiment Details

Bucket-restricted RLVR (Figure 5). We train variants restricting updates to a single bucket $B_{m/8}$ for $m \in \{0, 1, 4, 8\}$. Rollout outcomes are computed as usual, but only problems in the target bucket contribute to the gradient. We use the REINFORCE update with signed rewards $r \in \{-1, +1\}$ and the shared hyperparameters in Table 3. The signed reward gives $B_{0/8}$ a negative reinforcement signal; standard GRPO instead assigns zero normalized advantage to this uniform-outcome group.

Early stopping (Table 1). We select the checkpoint at epoch 5 or 10 from the standard 20-epoch REINFORCE run; no additional tuning is performed.

Bucket masking (Table 1). The masking variants (mask $B_{8/8}$, mask $B_{\geq 7/8}$, mask $B_{\geq 6/8}$) exclude problems in the specified high-success buckets from the gradient. Like BBG, these apply hard binary masks with equal problem-level weights among retained problems; the variants differ in which buckets are excluded.

Coverage tracking (Figure 4). A problem is classified as covered if it produces at least one correct response out of N=256 samples, and uncovered otherwise. We identify problems with Pass@256=0 under the base policy and track their status at each checkpoint. Entry and loss counts in Figure 4(b) are relative to the base classification at each checkpoint; the sampling-noise analysis in Appendix C addresses the concern that some transitions may be artifacts of finite-sample evaluation.

A.8 Reproducibility

All experiments use fixed random seeds for data shuffling. Rollout generation via vLLM is non-deterministic due to GPU parallelism. The reported comparisons do not quantify variation across independent training runs.

B Analysis for the Zero-Success Bucket

For a problem with true single-sample success probability p, its contribution to Pass@k is

$$J_k(p) = 1 - (1-p)^k. \tag{13}$$

The remaining headroom is

$$1 - J_k(p) = (1-p)^k, \tag{14}$$

and the marginal gain from increasing p is

$$\frac{dJ_k(p)}{dp} = k(1-p)^{k-1}. \tag{15}$$

Thus, J_k saturates exponentially fast in k. Further optimization on a problem may still increase Pass@1 by making correct responses more reliable, while offering little direct gain in high-k coverage. The derivatives compare the metric value of a unit increase in p; the actual change induced by a parameter update is a separate quantity.

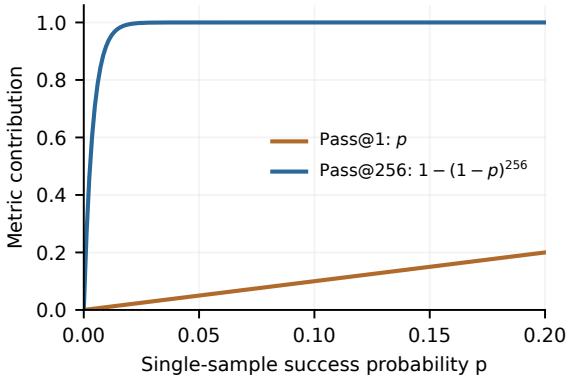


Figure 6: *Coverage saturates before single-attempt accuracy.* For a problem with true success probability p, Pass@1 equals p, while Pass@256 equals $1-(1-p)^{256}$. At $p = 0.05$, Pass@256 is already approximately 0.999998, despite substantial remaining Pass@1 headroom.

To account for uncertainty in p, we adopt the uniform prior used in Section 3.2. Observing m successes in n rollouts gives

$$p \mid m, n \sim \mathrm{Beta}(m+1, n-m+1). \tag{16}$$

The posterior expected marginal contribution is

$$u(m; n, k) = \mathbb{E}\big[k(1-p)^{k-1} \mid m, n\big] \tag{17}$$

$$= k\frac{\mathrm{B}(m+1, n-m+k)}{\mathrm{B}(m+1, n-m+1)} \tag{18}$$

$$= k\prod_{j=1}^{k-1}\frac{n-m+j}{n+1+j}, \tag{19}$$

where $\mathrm{B}(a,b)$ is the beta function. For adjacent buckets with $m < n$,

$$\frac{u(m+1; n, k)}{u(m; n, k)} = \prod_{j=1}^{k-1}\frac{n-m-1+j}{n-m+j} = \frac{n-m}{n-m+k-1}. \tag{20}$$

The product telescopes. For Pass@1, $u(m; n, 1) = 1$ in every bucket; for $k > 1$, the expected marginal utility decreases with each additional observed success.

In our main analysis, $k = 256$ and $n = 8$. The per-bucket utilities are

$$u(0; 8, 256) \approx 8.73, \qquad u(1; 8, 256) \approx 0.265, \qquad u(2; 8, 256) \approx 0.0071, \tag{21}$$

and

$$\frac{u(1;8,256)}{u(0;8,256)} = \frac{8}{263} \approx \frac{1}{32.9}. \tag{22}$$

Thus, under the same eight-rollout budget, the posterior expected marginal utility for Pass@256 in the one-success bucket is only about $1/33$ of that in the zero-success bucket. This is why $B_{0/n}$ is special: it has the largest expected direct coverage gain per unit increase in success probability, while that value falls sharply in higher-success buckets.

Under the uniform prior, the prior-predictive distribution of m is uniform over $\{0, \ldots, n\}$. The law of total expectation gives

$$\frac{1}{n+1} \sum_{m=0}^{n} u(m;n,k) = \int_0^1 k(1-p)^{k-1}\, dp = 1. \tag{23}$$

For $n = 8$ and $k = 256$, 96.97% of $\sum_m u(m;n,k)$ is concentrated on $B_{0/8}$. This normalization is over possible rollout observations under the prior; empirical bucket frequencies need not be uniform.

However, $B_{0/n}$ is also a noisy and tricky bucket. A problem can already have high Pass@256 while still producing zero successes in only $n = 8$ training rollouts. Indeed,

$$\Pr(\hat{p}_{\theta,n} = 0 \mid p) = (1-p)^n. \tag{24}$$

Writing $q = J_k(p) = 1 - (1-p)^k$, we can express this probability directly in terms of the Pass@k value:

$$\Pr(\hat{p}_{\theta,n} = 0 \mid J_k(p) = q) = (1-q)^{n/k}. \tag{25}$$

For $n = 8$ and $k = 256$,

$$\Pr(\hat{p}_{\theta,8} = 0 \mid J_{256}(p) = q) = (1-q)^{1/32}. \tag{26}$$

This probability remains large even when the problem is highly likely to be solved within 256 samples:

$J_{256}(p)$	$\Pr(\hat{p}_{\theta,8} = 0)$
0.50	97.9%
0.90	93.1%
0.99	86.6%
0.999	80.6%

(27)

Thus, $B_{0/8}$ should not be interpreted as a clean set of problems with low high-k coverage. It may also contain problems with low single-sample success probability but high coverage at the evaluation budget. This explains both sides of the intervention in Section 4.2: restricting updates to $B_{0/8}$ prioritizes the bucket with the largest posterior expected utility, but the bucket itself is still noisy because $n \ll k$.

One direct way to reduce this leakage is to increase the number of training rollouts n. For a problem with true success probability p, the probability that it falls into the zero-success bucket is

$$\Pr(\hat{p}_{\theta,n} = 0 \mid p) = (1-p)^n. \tag{28}$$

Thus, increasing n makes $B_{0/n}$ a cleaner proxy for problems with low single-sample success probabilities. For example, when $p = 0.05$, the leakage probability is $0.95^8 = 66.3\%$ under $n = 8$, but drops to $0.95^{16} = 44.0\%$ under $n = 16$, and further to $0.95^{32} = 19.4\%$ under $n = 32$. Since $p = 0.05$ already corresponds to $J_{256}(p) = 1 - 0.95^{256} \approx 0.999998$, these are precisely the kinds of high-coverage problems that can enter the zero-success bucket under small rollout budgets.

C Sampling-Noise Analysis

This appendix examines how much of the coverage entry and loss observed in Section 4.1 could be explained by finite-sample noise. Since Pass@k is estimated from a finite set of sampled completions (Chen et al., 2021), the induced binary Pass@256 coverage label is noisy at the problem level: a problem with a small but nonzero success probability may receive zero correct samples in one evaluation and at least one correct sample in another. We therefore distinguish weak one-success transitions from stronger multi-success transitions.

For problem i at checkpoint t, let $C_{i,t}$ be the number of correct responses among $K = 256$ independent evaluation samples. If the true single-sample success probability is $p_{i,t}$, then

$$\Pr(C_{i,t} = c) = \binom{K}{c} p_{i,t}^{c}(1 - p_{i,t})^{K-c}, \qquad c = 0, 1, \ldots, K. \tag{29}$$

The empirical Pass@256 coverage indicator is

$$Z_{i,t} = \mathbf{1}\{C_{i,t} > 0\}. \tag{30}$$

An observed coverage entry from the base model to checkpoint t is $C_{i,0} = 0$ and $C_{i,t} > 0$, while an observed loss is $C_{i,0} > 0$ and $C_{i,t} = 0$.

Exact noise test for an observed entry. Suppose a problem has $C_{i,0} = 0$ under the base model and $C_{i,t} = c$ at a later checkpoint. Under the null hypothesis that the true success probability did not change, $H_0 : p_{i,t} = p_{i,0}$, the c successes observed among the combined $2K = 512$ samples should be exchangeable between the two evaluations. Conditioning on the total number of successes, $C_{i,0} + C_{i,t} = c$, removes the unknown success probability. Under H_0, the c success positions are uniformly distributed among the $2K$ sample positions. Therefore, the probability that all c successes fall in the later checkpoint is

$$P_{\text{noise}}(c) = \Pr(C_{i,t} = c, C_{i,0} = 0 \mid C_{i,0} + C_{i,t} = c, H_0) = \frac{\binom{K}{c}}{\binom{2K}{c}}. \tag{31}$$

Equivalently, for $K = 256$,

$$P_{\text{noise}}(c) = \frac{\binom{256}{c}}{\binom{512}{c}} = \prod_{j=0}^{c-1} \frac{256 - j}{512 - j}. \tag{32}$$

This is the standard conditional form of Fisher's exact test for a 2×2 table (Fisher, 1922; Agresti, 2013). This gives the following exact one-sided sampling-noise scores:

c	1	2	3	4	5	6	7	10
$P_{\text{noise}}(c)$	0.5000	0.2495	0.1243	0.0618	0.0306	0.0152	0.0075	8.93×10^{-4}

Thus, a one-success entry is weak evidence and can easily arise from sampling noise. By contrast, entries with $c \geq 5$ have a conditional score below 0.05, and entries with $c \geq 7$ below 0.01. The same calculation applies to coverage loss. If $C_{i,0} = c$ and $C_{i,t} = 0$, then under the unchanged-success-probability null,

$$\Pr(C_{i,0} = c, C_{i,t} = 0 \mid C_{i,0} + C_{i,t} = c, H_0) = \frac{\binom{K}{c}}{\binom{2K}{c}}.$$

Therefore, losing a problem that had only one base correct sample is noisy, while a loss with many base correct samples has a smaller conditional noise score.

These scores are individual-comparison quantities: they do not account for selecting problems by their base evaluation or for comparisons across many problems and checkpoints. We use the counts below to describe the strength and persistence of observed transitions, rather than as a test of aggregate statistical significance.

Observed entry under standard RLVR. We apply this calculation to the coverage-tracking experiment in Section 4.1. On the held-out benchmarks AIME 2025, MinervaMath, and OlympiadBench, there are $15 + 57 + 193 = 265$ problems with $C_{i,0} = 0$ under the base model.

At epoch 2, $70/265 = 26.4\%$ of these base zero-success problems enter the empirical Pass@256 coverage set. Of these 70 entries, 27 have at least two correct samples and 3 have at least five correct samples; equivalently, these account for $27/265 = 10.2\%$ and $3/265 = 1.1\%$ of all base zero-success problems. For the two intermediate-difficulty benchmarks emphasized in the main text, MinervaMath has $18/57 = 31.6\%$ entries, of which 9 have $c \geq 2$ and 2 have $c \geq 5$. OlympiadBench has $47/193 = 24.4\%$ entries, of which 15 have $c \geq 2$ and 1 has $c \geq 5$. Hence, many binary entries are indeed weak one-success events, but a subset is supported by repeated correct samples. At the final checkpoint, epoch 20 (step 140), $38/265 = 14.3\%$ of the held-out base zero-success problems

have an observed success. Among these 38 entries, 17 have $c \geq 2$, 3 have $c \geq 5$, and one has $c \geq 10$; as fractions of all base zero-success problems, these are 6.4%, 1.1%, and 0.4%, respectively.

Across all six evaluated checkpoints, $134/265 = 50.6\%$ of the base zero-success held-out problems enter at least once. Among them, $64/265 = 24.2\%$ enter at least once with $c \geq 2$, $11/265 = 4.2\%$ enter at least once with $c \geq 5$, and $4/265 = 1.5\%$ enter at least once with $c \geq 7$. In addition, $68/265 = 25.7\%$ enter in at least two consecutive evaluated checkpoints. These counts distinguish repeated and multi-success transitions from isolated one-success entries.

Observed loss also includes multi-success problems. The aggregate Pass@256 decline reflects not only entry but also coverage loss. At epoch 20, the held-out benchmarks have 152 losses among $15 + 215 + 481 = 711$ problems with observed base-model successes, i.e. $152/711 = 21.4\%$. Many of these losses come from problems that had multiple correct samples under the base model: 91 had at least two base correct samples, and 32 had at least five base correct samples.

Overall, at epoch 20 the held-out benchmarks have 38 entries but 152 losses, giving $38 - 152 = -114$ net coverage transitions. Over the 976 held-out problems, this corresponds to $-114/976 = -11.7\%$. This matches the observed aggregate Pass@256 degradation: RLVR produces newly observed successes on some problems, but this gain is more than offset by losses on problems with previously observed successes.

D Additional Coverage Dynamics

Figure 7 gives the per-benchmark Pass@k curves, and Figure 8 reports coverage entry and loss.

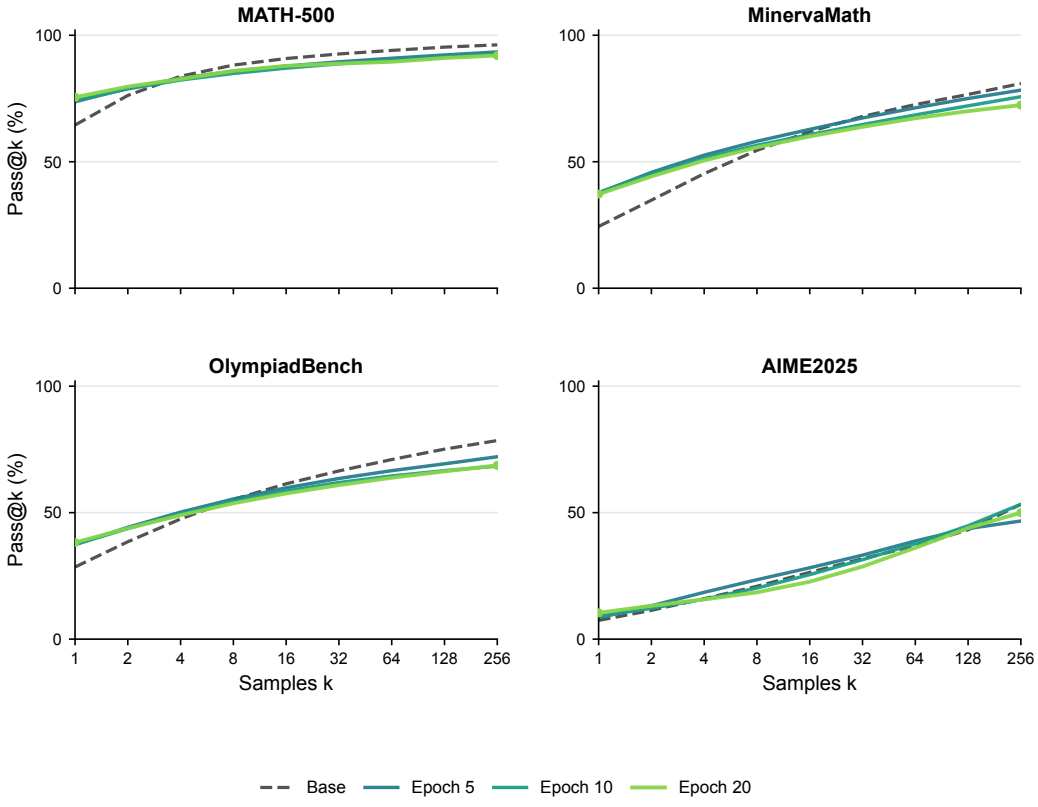


Figure 7: *Per-benchmark Pass@k curves across training.* The averaged curve in Figure 2(b) is decomposed by benchmark. On intermediate-difficulty benchmarks such as MinervaMath and OlympiadBench, later RLVR epochs improve or preserve low-k performance while reducing high-k Pass@k. On easier or harder benchmarks, this divergence is less pronounced within the evaluated range of k.

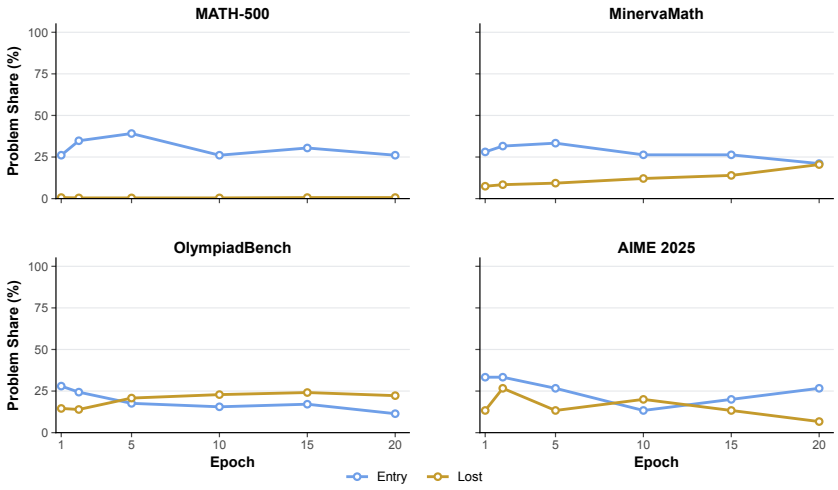


Figure 8: *Per-benchmark coverage entry and loss under standard RLVR.* Entry denotes the fraction of base zero-success problems with an observed success at a later checkpoint; loss denotes the fraction of base-covered problems with no later success. Each evaluation uses 256 samples. The two rates have different denominators.