Where Rollouts Begin: Low-Load, High-Leverage First-Token Diversification for RLVR

Anonymous Authors[1]

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) trains reasoning models without labeled trajectories, relying on grouped rollouts to expose the policy to alternative reasoning paths and a verifier to score them. Rollout diversity has accordingly emerged as a central bottleneck in RLVR, with most existing methods broadening exploration through temperature, prefix, or rollout-selection adjustments. We identify a structurally distinguished but overlooked position for broadening this diversity: the *first token* after the reasoning marker. The policy's first-token distribution exhibits a sharply peaked yet correctness-decoupled phenomenon, and this first token position can broaden the regions a rollout group covers without altering the correctness signal. We introduce REFT (Rollout Exploration with First-Token Diversification), a light addition to the RLVR pipeline that samples first tokens uniformly from the policy's own top-N candidates and allocates rollouts evenly, leaving every other component unchanged. Trained on the resulting diversified rollouts, REFT improves aggregate Pass@1, Pass@8, and Pass@64 over DAPO and GRPO baselines across four base models (0.5B-7B) and three difficulty regimes.

1. Introduction

Reinforcement learning with verifiable rewards (RLVR) has become a central recipe for improving the reasoning ability of language models without labeled trajectories (Guo et al., 2025; Hu et al., 2025). Instead of imitating annotated solutions, the policy samples candidate reasoning traces, an automatic verifier scores their final answers, and the model is updated from rewards. In grouped methods such as GRPO (Shao et al., 2024) and DAPO (Yu et al., 2025), this update is constructed from a finite set of rollouts for the same prompt. The rollout group is therefore the learner's window onto the space of possible reasoning paths. If all rollouts receive the same reward, the group provides little contrastive signal. If the rollouts are semantically redundant, the verifier can only compare a narrow set of alternatives.

This makes rollout diversity a central bottleneck in RLVR. Most existing approaches seek diversity where it appears most natural, such as high-entropy reasoning pivots (Wang et al., 2025; Yu et al., 2026; Wei et al., 2026), trajectory-level branches (Xing et al., 2026; Hou et al., 2025; Zhao et al., 2026; Hu et al., 2026; Wan et al., 2026), or final-answer alternatives (Song et al., 2025; Wu et al., 2025; Zhu et al., 2025). Other methods broaden exploration through temperature-controlled sampling (Zhuang et al., 2025; Yang et al., 2025; Liu et al., 2025b), while efficiency-oriented methods reduce rollout cost through reuse and selective sampling (Huang & Wan, 2026; Liu et al., 2025a; Chang et al., 2026; Chen et al., 2026). These methods differ in mechanism, but they share a practical assumption that early low-entropy prefix tokens are not where valuable exploration lives.

We challenge this assumption by focusing on the first generated token after the reasoning marker <think>. This token is usually a discourse opener rather than a substantive reasoning step, so it appears too minor to matter. We show the opposite. The first token is *low-load* in semantic content but *high-leverage* in distributional effect. Because it is the first autoregressive choice in the reasoning trace, changing it affects the conditional model for every subsequent token.

In our diagnostic, the first-token distribution is sharply concentrated, yet rollout correctness remains nearly flat across the top-20 alternatives. Even low-probability tokens within this set can lead to correct rollouts when the remaining continuation is sampled normally. Thus, these first tokens are not necessarily poor reasoning choices; they are undersampled routes into viable continuation regions. This routing effect is not confined to the opener itself. After stripping the first token, continuations generated from different first tokens occupy different semantic regions, and uniform top-20 first-token sampling produces substantially higher continuation diversity than standard sampling.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the FoGen Workshop at ICML 2026. Do not distribute.

This suggests a different use of a fixed rollout budget. Instead of injecting noise where the model is uncertain and correctness is fragile, we can diversify where the model is artificially certain and correctness remains comparatively stable. We introduce **REFT**: Rollout Exploration with First-Token Diversification. For each prompt, REFT takes the policy's own top-N valid first-token candidates, samples K uniformly, and allocates the rollout budget evenly across the selected first tokens. All continuations are then generated with the unchanged decoder. The verifier, reward, advantage estimator, RL objective, and rollout budgets are unchanged. Thus, REFT is not a new RL algorithm. It is a targeted replacement for the first decision made by the rollout sampler.

This targeted intervention also avoids the main weakness of high-temperature exploration. Raising temperature flattens every token distribution, including later reasoning steps where low-probability choices are often harmful, while still giving little control over whether rare but viable first tokens appear in a small rollout group. REFT instead guarantees first-token coverage under the same rollout budget and keeps the rest of the trajectory sampled exactly as in the baseline.

Empirically, under matched rollout budgets, REFT improves RLVR performance across four base models, two grouped-RL objectives, and three training datasets. Across Pass@k metrics with $k \in \{1, 8, 64\}$, REFT consistently outperforms GRPO/DAPO baselines on the various math benchmarks. Further analysis shows that these gains come from broader training-time continuation and answer coverage, fewer all-wrong groups, and reduced first-token overcrediting. Together, these results suggest a simple inversion of the usual exploration intuition. In RLVR, useful diversity can be recovered not only from high-entropy reasoning forks, but also from low-load prefix choices whose probability is sharply biased and weakly tied to correctness.

2. Preliminaries

RLVR and grouped rollouts. Reinforcement learning with verifiable rewards (RLVR) trains a language-model policy from sampled attempts rather than labeled reasoning trajectories. Given a prompt x, the policy samples a *rollout* y, including both the reasoning trace and the final answer, and an automatic verifier returns a scalar reward $R(x, y)$. Group-based RLVR methods sample G rollouts for the same prompt and update the policy from their relative rewards. In GRPO (Shao et al., 2024), the reward for rollout y_i is converted to a group-normalized advantage

$$A_i = \frac{R_i - \mathrm{mean}(\{R_j\}_{j=1}^{G})}{\mathrm{std}(\{R_j\}_{j=1}^{G}) + \epsilon}, \tag{1}$$

where $R_i = R(x, y_i)$. A PPO-style clipped objective then applies this trajectory-level advantage to the tokens of y_i. The finite rollout group is the unit from which the learning signal is constructed. Recent RLVR systems and variants build on this grouped-rollout template: DeepSeek-R1 (Guo et al., 2025) scales verifier-based reasoning RL, and DAPO (Yu et al., 2025) improves GRPO with dynamic sampling, Clip-Higher, token-level loss, and overlong-response handling. Recent methods further improve grouped-rollout data construction or usage. GRESO (Zheng et al., 2025) filters prompts, AR3PO (Zhang et al., 2025) reuses rollouts, RL-ZVP (Le et al., 2026) learns from zero-variance prompts, and PODS (Xu et al., 2025) selects informative rollout subsets for update.

Why rollout diversity matters. Equation (1) makes diversity a first-order concern. If all rollouts in a group receive the same reward, the group has no reward contrast and contributes little or no policy-gradient signal. Even with non-identical rewards, near-duplicate reasoning traces provide a narrow signal because the verifier can only compare alternatives that were actually sampled. Diverse rollouts therefore serve two roles. They increase the probability that a finite group contains useful reward variation, and broaden the set of reasoning regions over which the verifier can assign credit. This is particularly important for reasoning models evaluated by pass@k, where retaining multiple plausible solution modes can matter as much as increasing the probability of the modal response. Recent analyses of outcome-level diversity collapse (Song et al., 2025), support narrowing under RLVR (Wu et al., 2025), and sample-reinforcement dynamics (Zhu et al., 2025) point to a common concern that improving Pass@1 can narrow the accessible rollout distribution and hurt exploration or inference-time scaling.

Existing routes to rollout diversity. Most prior diversity methods allocate rollout budget to positions that appear semantically or statistically important. Entropy-based methods focus on high-uncertainty pivots or token-level entropy regulation (Wang et al., 2025; Yu et al., 2026), and Entropy-Tree branches the decoding process at high-entropy positions (Wei et al., 2026). Trajectory- and tree-level methods explicitly structure rollout exploration. LATR branches at high-uncertainty steps and prunes non-divergent continuations (Xing et al., 2026), TreeRL incorporates on-policy tree search (Hou et al., 2025), and ROSE branches by semantic entropy (Zhao et al., 2026). Other trajectory-level methods add global or multi-scale diversity incentives to encourage different derivations (Hu et al., 2026; Wan et al., 2026). Outcome-level methods and analyses study final-answer diversity or sample-reinforcement dynamics (Song et al., 2025; Zhu et al., 2025). Temperature-based methods control exploration through temperature schedules or annealed decoding (Zhuang et al., 2025; Yang et al., 2025), while residual-prompt ERPO increases exploration on all-correct prompts to recover training signal (Liu et al., 2025b). A separate line improves rollout efficiency through prefix sharing (Huang & Wan, 2026), speculative continuations (Liu et al., 2025a), tree-structured caches (Chang et al., 2026), or budgeted rejection (Chen et al., 2026). Although these approaches differ, they share a practical bias that early prefix choices are treated as fixed scaffolding, shared for efficiency, or bypassed in favor of later reasoning decisions. We focus on the first generated token after the reasoning marker <think>, a discourse opener that prior diversity methods treat as fixed scaffolding to be shared or bypassed.

3. Diagnosis: Low-Load, High-Leverage First Tokens

We define the *first token* as the first valid semantic token after the reasoning marker <think>, ignoring whitespace-only tokens and line breaks. At first glance, this position looks like a poor place to spend rollout budget. In math reasoning traces, the first token is usually a discourse opener such as "First", "Let", or "To" (Appendix E); it rarely performs algebra, chooses a case, assigns a variable, or commits to an answer. This section shows that this intuition is incomplete. The first token has low task-specific load, but it has unusually high distributional leverage. The diagnostics in this section are conducted with Qwen2.5-3B-Instruct (Yang et al., 2024) on GSM8K (Cobbe et al., 2021).

The policy is confident, but the verifier is not. For each prompt x, let $f_r(x)$ be the rank-r token under the policy distribution at the first token position,

$$\pi_\theta(\cdot \mid x, \texttt{<think>}).$$

The model's prior over this position is extremely sharp. Figure 1 shows that the top-ranked first token receives mean probability 0.57 at temperature $T = 1.0$. Even at $T = 2.0$, the top-ranked token still receives probability 0.38 on average. Thus, before any substantive reasoning step has been generated, standard sampling has biased the rollout group toward a small set of openers. Appendix E provides the same distributions for other model-dataset pairs.

The surprising part is that this probability ranking is only weakly aligned with correctness. We force the first token to be $f_r(x)$ and then sample the remaining continuation with the unchanged decoder. As shown by the red curve in Figure 1, rollout correctness is nearly flat across the top-20 ranks. The rank-20 first token has probability only 2.68×10^{-5}, yet achieves 70.40% correctness, compared with 75.29% for the rank-1 token.

This is the central empirical asymmetry: at the first-token position, model probability and verifier correctness decouple. A low-probability first token is not necessarily a bad reasoning choice. Often, it is merely an under-sampled way of beginning a still-viable solution.

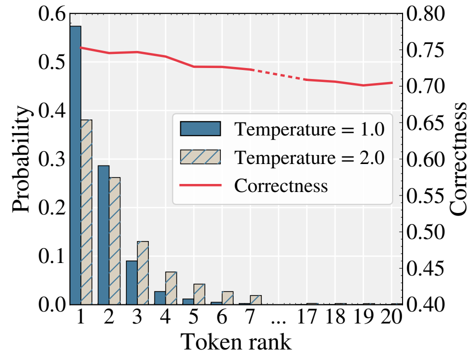


Figure 1. **Sharp probability, flat correctness.** The model assigns high probability to the top first token, but rollout correctness remains nearly flat across the top-20 ranks.

The first token matters not because of what it says locally, but because of where it routes the rollout. If a response is decomposed into the first token F and continuation z, then

$$\pi_\theta(F, z \mid x, \texttt{<think>}) = \pi_\theta(F \mid x, \texttt{<think>}) \cdot \pi_\theta(z \mid x, \texttt{<think>}, F). \tag{2}$$

Varying F therefore induces distinct continuation distributions over the rest of the rollout. A token with tiny marginal probability can therefore induce a competent but rarely visited continuation distribution.

A rare first token changes the continuation model. To test whether this routing effect is real, we compare four rollout strategies with the same group size: standard sampling, forcing $f_1(x)$, forcing $f_{20}(x)$, and sampling uniformly from the top-20 first tokens. We then remove the first token from every rollout and measure semantic diversity among the remaining continuations (details in Appendix D).

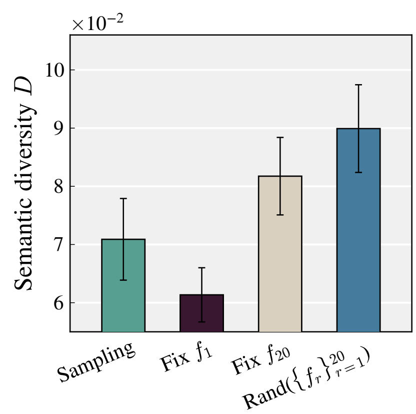


Figure 2. **First tokens route continuations.** Semantic diversity is measured after stripping the first token itself, so the effect comes from the rest of the rollout.

Figure 2 shows that first-token choice changes the continuation distribution. Forcing a rare first token already increases diversity over standard sampling, and uniform top-20 sampling yields the largest effect. The result is important for three reasons. First, the diversity is measured after removing the opener, so it is not just surface variation in the first word. Second, even a fixed rank-20 opener produces more diverse continuations than sampling. Third, sampling uniformly across the top-20 first tokens yields the highest diversity of all four strategies, suggesting that the contributions from different first tokens are complementary rather than redundant. The first token is therefore a routing variable. It can expose continuation regions that standard rollouts almost never visit, while preserving comparable correctness.

GRPO sharpens the wrong preference. The bias is not only present before RLVR training; GRPO-style updates can amplify it. A trajectory-level advantage is applied to every token in the rollout. For a rollout whose first token is F_i, the update contains the term

$$A_i \nabla_\theta \log \pi_\theta(F_i \mid x, \texttt{<think>}),$$

where A_i is the group-normalized advantage from Equation (1). This assigns the same scalar credit to the generic opener as to the later tokens that actually solve the problem.

Because the top-ranked first token is sampled much more often, it also receives many more chances to co-occur with positive-advantage trajectories. Rare first tokens, even when they would lead to correct or diverse continuations, may not appear in the group at all. As Figure 3 shows, the top-1 first-token probability grows during training. In this sense, standard GRPO can over-credit a discourse preference that the verifier does not strongly support.

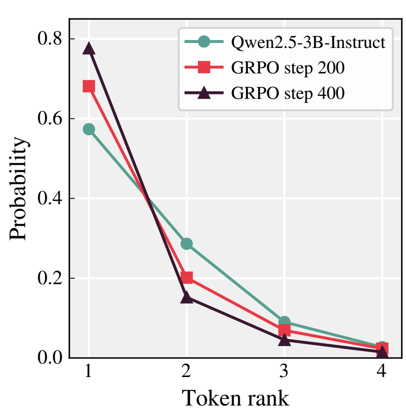


Figure 3. **RLVR sharpens the first-token prior.** The top-1 first-token probability increases during training, although the first token is only weakly tied to correctness.

Why temperature is not the answer. A natural treatment is to increase sampling temperature. However, temperature is too blunt for this problem. It perturbs every token distribution in the rollout, including later reasoning steps where low-probability choices are much more likely to be harmful. It also still fails to cover rare first tokens under a small rollout budget. As shown in Figure 1, the first-token prior remains sharp even at $T = 2.0$: the rank-20 first token has probability only 1.96×10^{-3}. With group size $G = 8$, the probability of sampling it at least once is $\sim 1.6\%$. Thus, temperature injects noise throughout the continuation while offering little control over first-token coverage.

4. REFT: Rollout Exploration with First-Token Diversification

The diagnosis in Section 3 suggests a simple opportunity. The first token after <think> is *low-load*: changing it usually does not directly change the mathematical content of the solution. It is *correctness-stable*: low-probability top-ranked openers can still lead to correct rollouts. And it is *high-leverage*: once chosen, it conditions the entire continuation. Thus, the first token is an unexpectedly cheap exploration site. Instead of spending rollout budget where the model is uncertain and correctness is fragile, we diversify where the model is artificially certain and correctness remains stable.

REFT (Rollout Exploration with First-Token Diversification) turns this observation into a minimal rollout intervention. It modifies only the first-token sampling. The verifier, advantage estimator, training objective, model architecture, continuation decoder, and rollout budget are unchanged.

Candidate set. For each prompt x, we compute the policy distribution at the first-token position and keep the top-N valid candidates:

$$\mathcal{F}_N(x) = \{f_1(x), \ldots, f_N(x)\},$$
$$\pi_{\theta_{\text{old}}}(f_1 \mid x, \texttt{<think>}) \geq \cdots \geq \pi_{\theta_{\text{old}}}(f_N \mid x, \texttt{<think>}).$$

Here $f_1, \ldots, f_N$ are the highest-probability valid first tokens under the behavior policy. We exclude whitespace-only and special tokens. Because the candidates come from the policy's own top-N list, REFT does not introduce external prompts or hand-written prefixes; it reallocates budget among first tokens the model already considers plausible.

First-token diversification. Standard rollout sampling draws all G rollouts from the original first-token distribution, which is often dominated by a single opener. REFT instead stratifies the rollout group by first token. Given the top-N candidate set $\mathcal{F}_N(x)$, we sample K distinct first tokens uniformly without replacement:

$$\mathcal{S}_K(x) \sim \text{Unif}\left(\{\mathcal{S} \subseteq \mathcal{F}_N(x) : |\mathcal{S}| = K\}\right).$$

We then allocate G/K rollouts to each selected token. For every $f \in \mathcal{S}_K(x)$, REFT fixes f as the first token and samples the continuation with the unchanged baseline decoder:

$$z_{f,j} \sim \pi_{\theta_{\text{old}}}\left(\cdot \mid x, \texttt{<think>}, f\right), \qquad j = 1, \ldots, G/K.$$

Algorithm 1 REFT rollout sampler

Require: Prompt x; behavior policy $\pi_{\theta_{\text{old}}}$; group size G; candidate width N; selected-token count K
Ensure: Rollout group $\mathcal{Y}_G^{\text{REFT}}(x)$
1: $\mathcal{F}_N(x) \leftarrow \{f_1(x), \ldots, f_N(x)\}$ {Top-N valid first tokens}
2: Sample $\mathcal{S}_K(x) \subseteq \mathcal{F}_N(x)$ uniformly with $|\mathcal{S}_K(x)| = K$
3: **for** each $f \in \mathcal{S}_K(x)$ **do**
4: Sample $z_{f,1}, \ldots, z_{f,G/K} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x, \texttt{<think>}, f)$
5: **end for**
6: $\mathcal{Y}_G^{\text{REFT}}(x) \leftarrow \{(f, z_{f,j}) : f \in \mathcal{S}_K(x),\ j = 1, \ldots, G/K\}$
7: **return** $\mathcal{Y}_G^{\text{REFT}}(x)$

The resulting rollout group is

$$\mathcal{Y}_G^{\text{REFT}}(x) = \{(f, z_{f,j}) : f \in \mathcal{S}_K(x),\ j = 1, \ldots, G/K\}.$$

Thus, REFT replaces a sharply peaked first-token draw with explicit coverage of K plausible openers, while leaving all subsequent generation unchanged.

What remains unchanged. After constructing $\mathcal{Y}_G^{\text{REFT}}(x)$, training proceeds exactly as in the underlying RLVR algorithm. The verifier scores each rollout, group-normalized advantages are computed as usual, and the same GRPO/DAPO-style objective is applied to the sampled trajectories. REFT is therefore not a new RL objective. It is a drop-in replacement for the rollout sampler's first decision, and it can be layered under existing refinements such as dynamic sampling, asymmetric clipping, token-level loss, or overlong-response handling.

Relation to temperature. REFT targets the failure mode identified in Section 3. Increasing temperature flattens every token distribution in the rollout, including later reasoning steps where low-probability choices are more likely to be harmful. It also gives little control over which first-token alternatives actually appear in a small rollout group. REFT instead controls coverage at the first-token position while leaving the continuation decoder unchanged. It broadens the rollout group where correctness is comparatively stable, without injecting global noise into the reasoning trace.

Rollout construction with a fixed RL objective. REFT changes the data-collection distribution only at the first-token position. We train the completed rollouts with the same GRPO/DAPO objective used by the baselines, without an additional importance correction for the first-token sampler. This keeps the downstream RL objective fixed while isolating the effect of rollout construction. We do not claim that this gives an unbiased estimator of the standard on-policy rollout objective. Rather, REFT replaces a faithful draw from the sharply peaked first-token prior with a controlled allocation of the same finite rollout budget across several plausible first tokens. Details are in Appendix B.

5. Experiments

Setup. We compare DAPO (Yu et al., 2025) and GRPO (Shao et al., 2024) with their REFT-augmented variants, where REFT only modifies rollout sampling. Experiments use four instruction-tuned base models across three scales: Qwen2.5-0.5B/3B/7B-Instruct (Yang et al., 2024) and Llama3.2-3B-Instruct (Grattafiori et al., 2024). We train on GSM8K (Cobbe et al., 2021), BigMath-Easy, and BigMath (Albalak et al., 2025), spanning elementary to harder math-reasoning regimes. GSM8K uses rollout group size $G = 8$, while BigMath-Easy and BigMath use $G = 16$. Unless otherwise stated, REFT uses candidate width $N = 20$ and selected-token count $K = 4$, allocating rollout slots evenly across selected first tokens. All methods use matched rollout budgets and the same rollout decoding configuration. Appendix A.2 reports no measurable net runtime overhead.

GSM8K-trained models are evaluated on GSM8K (Cobbe et al., 2021), while BigMath- and BigMath-Easy-trained models are evaluated on MATH-500 (Lightman et al., 2024), AIME 2024 (Zhang & Math-AI, 2024), AIME 2025 (Zhang & Math-AI, 2025), and AMC 2023 (Li et al., 2024). We report Math Avg. as the unweighted average over MATH-500, AIME 2024, AIME 2025, and AMC 2023, excluding GSM8K. Inference uses vLLM (Kwon et al., 2023) with temperature 0.6 and top-$p = 0.95$, and we report Pass@1, Pass@8, and Pass@64. Additional training, reward, inference, hardware, and dataset details are in Appendix A.1.

Consistent gains across models and sampling budgets. Table 1 reports Pass@$1/8$ across three base models, two RLVR algorithms, and five benchmarks. On GSM8K, where the models start from strong baselines, REFT consistently improves performance, and the gains become larger on Math Avg., showing that REFT is especially effective when additional rollout exploration matters more. The distinction between Pass@1 and 8 is important for our method. Pass@1 measures individual sample quality, whereas Pass@8 captures finite-budget coverage: whether at least one correct reasoning path appears in the rollout group. The simultaneous gains show that the additional first-token coverage does not come at the expense of sample-level correctness. Rather than only spreading samples across more openings, REFT improves both individual completion accuracy and the probability that a rollout group contains a correct trajectory.

Gains hold at the smallest scale. Table 2 shows that the effect of REFT is not limited to larger models. Even at 0.5B, where reasoning capacity is limited, REFT improves both GRPO and DAPO across Pass@$1/8/64$ on GSM8K. Thus, first-token diversification is not contingent on large model capacity, and the same low-load, high-leverage position remains effective across scales from 0.5B to 7B parameters.

Table 1. Reasoning performance across models and RLVR methods. Pass@1 denotes average single-sample correctness, and Pass@8 denotes whether at least one of 8 sampled completions is correct.

	GSM8K		MATH-500		AIME24		AIME25		AMC23		Math Avg.	
Method	Pass@1	Pass@8	Pass@1	Pass@8	Pass@1	Pass@8	Pass@1	Pass@8	Pass@1	Pass@8	Pass@1	Pass@8
Qwen2.5-3B-Instruct	80.14	94.16	51.40	78.00	6.67	16.67	0.00	13.33	35.00	75.00	23.27	45.75
GRPO	84.53	95.38	59.80	79.40	10.00	20.00	3.33	16.67	42.50	77.50	28.91	48.39
GRPO + REFT	**86.28**	**96.13**	**61.00**	**81.20**	**13.33**	**23.33**	**6.67**	**20.00**	**47.50**	**85.00**	**32.13**	**52.38**
DAPO	86.05	95.15	60.00	78.60	10.00	20.00	6.67	16.67	45.00	80.00	30.42	48.82
DAPO + REFT	**86.20**	**96.06**	**61.20**	**80.00**	**13.33**	**26.67**	**13.33**	**20.00**	**52.50**	**85.00**	**35.09**	**52.92**
Llama3.2-3B-Instruct	73.24	91.96	29.80	64.00	3.33	13.33	0.00	3.33	20.00	50.00	13.28	32.66
GRPO	75.66	92.12	35.60	64.80	6.67	16.67	3.33	3.33	22.50	60.00	17.02	36.20
GRPO + REFT	**77.63**	**94.09**	**36.80**	**65.40**	6.67	**20.00**	3.33	**6.67**	**25.00**	**62.50**	**17.95**	**38.64**
DAPO	73.84	93.10	37.20	65.60	10.00	23.33	3.33	6.67	22.50	60.00	18.26	38.90
DAPO + REFT	**78.47**	**94.01**	**41.00**	**67.00**	**13.33**	23.33	**6.67**	**10.00**	**25.00**	**65.00**	**21.50**	**41.33**
Qwen2.5-7B-Instruct	88.48	96.74	53.40	80.00	10.00	20.00	0.00	23.33	37.50	75.00	25.23	49.58
GRPO	91.05	97.12	72.00	86.40	13.33	23.33	10.00	26.67	55.00	85.00	37.58	55.35
GRPO + REFT	**91.89**	**97.42**	**73.60**	**87.20**	**20.00**	**30.00**	**16.67**	**36.67**	**65.00**	85.00	**43.82**	**59.72**
DAPO	91.36	96.74	72.40	86.40	16.67	26.67	10.00	30.00	57.50	80.00	39.14	55.77
DAPO + REFT	**92.27**	**97.27**	**73.00**	**86.80**	**20.00**	**30.00**	**13.33**	**33.33**	**62.50**	**85.00**	**42.21**	**58.78**

Table 2. GSM8K performance of Qwen2.5-0.5B-Instruct.

	GSM8K		
Method	Pass@1	Pass@8	Pass@64
Qwen2.5-0.5B-Instruct	43.37	77.94	90.67
GRPO	51.71	78.17	91.05
GRPO + REFT	**54.21**	**79.68**	**93.71**
DAPO	52.14	78.92	92.80
DAPO + REFT	**53.62**	**79.45**	**94.39**

Pass@64 shows that the gains persist under a larger budget. RLVR can improve Pass@1 by concentrating probability on a narrower set of trajectories, potentially reducing discovery under larger sampling budgets (Yue et al., 2025). Table 3 reports Pass@64 on GSM8K and Math Avg.; full per-benchmark results are in Appendix C. On GSM8K, Pass@64 is near ceiling, so differences are necessarily small. Averaged across the six main model-objective pairs, REFT changes GSM8K Pass@64 by +0.59 points. The effect is clearer on Math Avg., where the same average reaches +3.13 points. REFT's Pass@1 gains do not come at the cost of larger-budget recoverability. The trained policy improves both at single-sample correctness and at finding at least one correct trajectory when more samples are available.

Candidate width and within-token replication. We further verify that REFT is robust to variations in candidate width Top-N and selected-token count K. Tables 4a and 4b report ablations on GSM8K with Qwen2.5-3B-Instruct, where the GRPO baseline achieves 84.53 Pass@1 and 95.38 Pass@8. Most REFT variants improve over the baseline, confirming robustness to these hyperparameter choices. Expanding the pool to Top-50 or Top-100 remains competitive in Pass@1 but gradually reduces Pass@8, suggesting that an overly wide candidate pool can dilute the finite rollout budget across too many openings.

Table 3. Pass@64 performance on GSM8K and Math Avg., measuring whether any of 64 sampled completions is correct.

	GSM8K	Math Avg.
Method	Pass@64	Pass@64
Qwen2.5-3B-Instruct	98.18	61.95
GRPO	98.64	64.34
GRPO + REFT	**99.39**	**67.82**
DAPO	97.65	65.13
DAPO + REFT	**98.18**	**69.18**
Llama3.2-3B-Instruct	97.42	51.61
GRPO	97.95	55.03
GRPO + REFT	**98.56**	**58.99**
DAPO	97.35	56.28
DAPO + REFT	**98.48**	**59.34**
Qwen2.5-7B-Instruct	97.95	66.26
GRPO	98.33	68.95
GRPO + REFT	**98.94**	**71.54**
DAPO	**98.48**	69.74
DAPO + REFT	98.41	**71.39**

Table 4. Ablation studies on candidate width Top-N and selected-token count K. The GRPO entry reports Qwen2.5-3B-Instruct trained with GRPO on GSM8K.

(a) Top-N, $K = 4$.

	GSM8K	
N	Pass@1	Pass@8
GRPO	84.53	95.38
10	86.20	95.83
20	**86.28**	**96.13**
50	85.82	95.30
100	85.14	94.92

(b) K, Top-$N = 20$.

	GSM8K	
K	Pass@1	Pass@8
GRPO	84.53	95.38
1	84.23	95.15
2	85.51	95.68
4	**86.28**	**96.13**
8	85.82	95.38

Fixing Top-$N = 20$, increasing K from 1 to 4 raises Pass@1 / Pass@8, showing the benefit of covering multiple first token regions within the same group. Increasing K further to 8 reduces the gain. With $G = 8$, this allocates only one continuation per selected first token, so a viable opening can be judged from a single unlucky trace. Our default $K = 4$ assigns two continuations per first token, balancing broader first-token coverage with enough replication to stabilize the rollout signal.

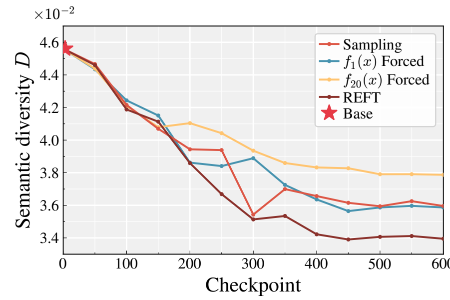
6. Analysis: When Diversity Becomes Useful

We now examine whether the diversity induced by REFT translates into useful RLVR training signal. We first study three group-level diagnostics: semantic diversity, outcome diversity within rollout groups, and zero-variance decomposition, then examine whether REFT also mitigates first-token over-crediting during training.

Training diversity need not remain high at inference. Figure 4a shows that REFT produces the most semantically diverse rollout groups during training, compared with standard sampling, forced $f_1(x)$, and forced $f_{20}(x)$. This matches the intended effect of first-token diversification, which exposes each rollout group to multiple early continuation regions instead of repeatedly following the most likely token. At inference time, however, Figure 4b shows that samples from the REFT-trained checkpoint are more concentrated. This is not contradictory. RLVR ultimately optimizes correctness, not semantic diversity itself. Read together with Tables 1 and 3, the lower inference-time diversity suggests that broader training-time exploration can lead to a policy that concentrates more strongly on successful continuation patterns, while still improving both sample-level accuracy and larger-budget coverage. Thus, the role of first-token diversity is to improve what the policy sees during training, not necessarily to make the final policy more diverse under standard inference sampling.

Outcome diversity beyond trajectory diversity. The near-flat first-token-conditioned accuracy in Section 3 shows viability, not equivalence. Plausible alternative first tokens are not destructive to correctness on average, but they can still route generation into different continuation regions. We therefore complement trajectory-level semantic diversity with final-answer diversity, a tractable proxy for outcome-space coverage used in prior work on reasoning diversity (Song et al., 2025; Dang et al., 2025). Figure 5 shows that REFT produces more distinct answers than standard sampling during training, suggesting that first token diversification changes not only the reasoning text but also the set of solution attempts under the same rollout budget. More distinct answers, however, do not by themselves imply useful exploration. A rollout group can contain many different wrong answers and still receive identical zero rewards, providing no contrast for learning. We therefore treat final-answer diversity as a coverage diagnostic and examine whether this broader coverage changes the correct-count distribution within each rollout group.

(a) Training-time rollout diversity

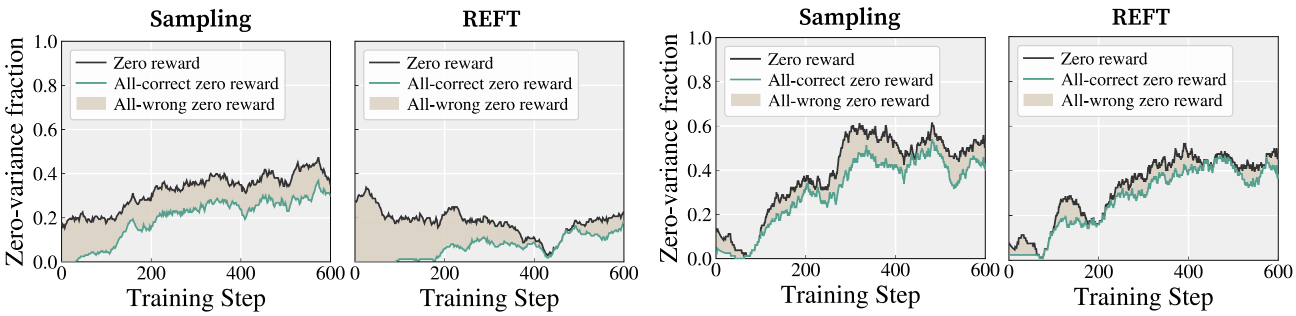


(b) Inference-time semantic diversity

Figure 4. **Sharper outputs without sacrificing coverage.** (a) During training, REFT produces the most semantically diverse rollouts among the four sampling strategies. (b) After training, the inference-time ordering inverts: REFT-trained checkpoints produce the most concentrated outputs, while Tables 1 and 3 show that this concentration coexists with stronger Pass@1 and Pass@64.

Figure 5. **Within-group answer diversity.** Number of unique final answers per rollout group during training on Qwen2.5-0.5B-Instruct, GSM8K, DAPO.

Zero-variance decomposition. With binary rewards, a zero-variance group can arise in two opposite ways. In an all-wrong group, every completion is incorrect, so the rollout budget fails to find any correct trajectory. In an all-correct group, every completion solves the prompt, so it is already saturated under the sampled policy. Aggregating these into a single zero-variance fraction collapses two opposite training



(a) Qwen2.5-0.5B-Instruct, GSM8K, DAPO

(b) Qwen2.5-3B-Instruct, BigMath-Easy, GRPO

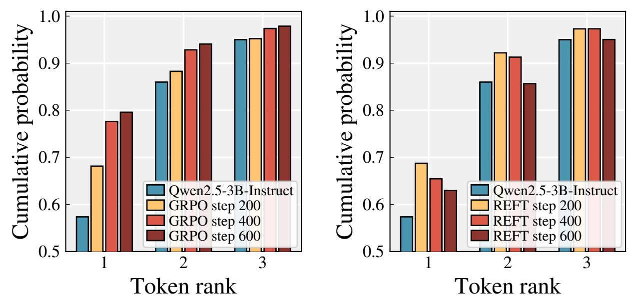
Figure 6. **REFT reduces all-wrong groups, the informative form of zero-variance reduction.** Each panel shows the fraction of zero-variance groups (black) and the all-correct subset (green) under standard sampling (left) and REFT (right). The shaded gap is the all-wrong subset.

conditions into one number, so the composition of the zero-variance set carries more information than its overall size.

Figures 6a and 6b show that REFT improves this composition. In Figure 6a, REFT initially exposes more incorrect trajectories as part of exploration, but the all-wrong area shrinks as training proceeds. This suggests that broader outcome coverage increasingly includes successful traces. In Figure 6b, REFT and standard sampling have relatively similar total zero-variance fractions, but REFT has a smaller all-wrong component. Thus, even when the overall zero-variance rate is comparable, fewer rollout groups fail to find any correct trajectory. Together with the increase in distinct final answers, this suggests that REFT is not merely scattering attempts across different wrong answers. Instead, REFT broadens the set of attempted solutions in a way that more often places at least one correct completion in the group, which is aligned with Pass@k metrics such as Pass@8.

REFT mitigates first-token over-crediting. Section 3 showed that standard GRPO assigns a biased preference and monotonically escalates it during training. Under standard sampling, the top-1 first-token probability keeps increasing throughout training (Figure 7a), while under REFT it remains comparatively flat (Figure 7b). This occurs because REFT reduces repeated visits to the same top-1 opener in a rollout group, reallocating the finite rollout budget across K plausible candidates from the policy's own top-N list.

Taken together. These analyses support a consistent picture. REFT uses first-token diversification to expose broader continuation regions and a wider set of attempted answers during training. The zero-variance decomposition shows that this coverage is useful rather than noisy, since fewer groups remain all-wrong. Figure 7 shows that REFT also reduces top-1 first-token over-sharpening. Overall, REFT reallocates the same rollout budget away from repeated visits to a weakly supported discourse preference and toward plausible first-token routes that more often yield useful reward contrast, consistent with gains in Pass@$1/8/64$.



(a) Standard GRPO

(b) GRPO + REFT

Figure 7. **REFT does not sharpen what the verifier ignores.** Top-r first-token probability over training on Qwen2.5-3B + GSM8K. Standard GRPO escalates the prior on the most-frequent first token; REFT keeps the prior comparatively flat.

7. Conclusion

We studied where a finite rollout budget should be diversified in RLVR. We focus on the first semantic token after the reasoning marker, which is a low-load choice upstream of the full continuation. Our diagnosis shows that the policy places a sharply concentrated prior on this position, although first-token rank is only weakly tied to downstream correctness. We introduced **REFT**, a minimal rollout-construction method that reallocates the first-token budget across plausible candidates from the policy's own top-N list. Because REFT changes only rollout construction, it isolates the effect of first-token coverage from changes to the verifier, objective, model, or decoder. Across model scales, datasets, and GRPO/DAPO objectives, REFT improves Pass@1, Pass@8, and Pass@64. The gains across Pass@k suggest that first-token coverage improves both single-sample quality and finite-budget recoverability. Analysis shows that these gains reflect useful diversity. REFT broadens continuation and answer coverage, reduces all-wrong zero-variance groups, and mitigates over-sharpening of the original top-1 first token. These results suggest that sharply biased, correctness-stable prefix choices can serve as effective intervention points for RLVR exploration.

Impact Statement

This work improves training efficiency for mathematical reasoning models, lowering compute costs and aiding educational tools. However, stronger reasoning capabilities carry dual-use risks, such as academic dishonesty or unsafe automation. While REFT simply optimizes existing pipelines, models trained with it should be deployed responsibly alongside robust safety evaluations, safeguards, and human oversight.

References

Albalak, A., Phung, D., Lile, N., Rafailov, R., Gandhi, K., Castricato, L., Singh, A., Blagden, C., Xiang, V., Mahan, D., et al. Big-math: A large-scale, high-quality math dataset for reinforcement learning in language models. *arXiv preprint arXiv:2502.17387*, 2025.

Bai, B., Wang, X., Ye, P., and Chen, T. Learning to explore with parameter-space noise: A deep dive into parameter-space noise for reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2602.02555*, 2026.

Chang, C.-C., Zhu, S., Zeng, Z., Lin, H., You, J., Abdelfattah, M. S., Jiang, Z., and Qian, X. Srt: Accelerating reinforcement learning via speculative rollout with tree-structured cache. *arXiv preprint arXiv:2601.09083*, 2026.

Chen, Z., Liu, H., Zhou, Y., Zheng, H., and Chen, B. Jackpot: Optimal budgeted rejection sampling for extreme actor-policy mismatch reinforcement learning. *arXiv preprint arXiv:2602.06107*, 2026.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Dang, X., Baek, C., Kolter, J. Z., and Raghunathan, A. Assessing diversity collapse in reasoning. In *ICLR 2025 Workshop on SSI-FM*, 2025.

Fang, Y., Lin, J., Fu, X., Qin, C., Shi, H., Hu, C., Pan, L., Zeng, K., and Cai, X. How to allocate, how to learn? dynamic rollout allocation and advantage modulation for policy optimization. *arXiv preprint arXiv:2602.19208*, 2026.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Gu, H., Wang, H., Liu, J., Li, L., Zhu, Q., Liu, B., Xu, B., Wang, L., Yang, X., Lin, S., et al. Qarl: Rollout-aligned quantization-aware rl for fast and stable training under training–inference mismatch. *arXiv preprint arXiv:2604.07853*, 2026.

Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., Xu, R., Zhang, R., Ma, S., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Hou, Z., Hu, Z., Li, Y., Lu, R., Tang, J., and Dong, Y. TreeRL: LLM reinforcement learning with on-policy tree search. In *ACL*, 2025.

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022.

Hu, J., Zhang, Y., Han, Q., Jiang, D., Zhang, X., and Shum, H.-Y. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. In *NeurIPS*, 2025.

Hu, Z., Zhang, S., Li, Y., Yan, J., Hu, X., Cui, L., Qu, X., Chen, C., Cheng, Y., and Wang, Z. Diversity-incentivized exploration for versatile reasoning. In *ICLR*, 2026.

Huang, B. and Wan, X. Pros: Towards compute-efficient rlvr via rollout prefix reuse. In *ICLR*, 2026.

Huang, W., Ge, Y., Yang, S., Xiao, Y., Mao, H., Lin, Y., Ye, H., Liu, S., Cheung, K. C., Yin, H., et al. Qerl: Beyond efficiency–quantization-enhanced reinforcement learning for llms. In *ICLR*, 2026.

Karouzos, C., Tan, X., and Aletras, N. Where does output diversity collapse in post-training? *arXiv preprint arXiv:2604.16027*, 2026.

Kirk, R., Mediratta, I., Nalmpantis, C., Luketina, J., Hambro, E., Grefenstette, E., and Raileanu, R. Understanding the effects of rlhf on llm generalisation and diversity. In *ICLR*, 2024.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *SOSP*, 2023.

Le, T.-L. V., Jeon, M., Vu, K., Lai, V., and Yang, E. No prompt left behind: Exploiting zero-variance prompts in llm reinforcement learning via entropy-guided advantage shaping. In *ICLR*, 2026.

Li, J., Beeching, E., Tunstall, L., Lipkin, B., Soletskyi, R., Huang, S., Rasul, K., Yu, L., Jiang, A. Q., Shen, Z., et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. https://huggingface.co/datasets/AI-MO/NuminaMath-TIR, 2024. Hugging Face dataset repository.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In *ICLR*, 2024.

Liu, B., Wang, A., Min, Z., Yao, L., Zhang, H., Liu, Y., Zeng, A., and Su, J. Spec-rl: Accelerating on-policy reinforcement learning via speculative rollouts. *arXiv preprint arXiv:2509.23232*, 2025a.

Liu, C., Liang, J., Jia, Y., Cao, B., Bai, Y., Huang, H., and Chen, X. Explore data left behind in reinforcement learning for reasoning language models. *arXiv preprint arXiv:2511.04800*, 2025b.

Rajbhandari, S., Rasley, J., Ruwase, O., and He, Y. Zero: memory optimizations toward training trillion parameter models. In *SC20*, 2020.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, 2019.

Setlur, A., Wang, Z., Cohen, A., Rashidinejad, P., and Xie, S. M. Reuse your flops: Scaling rl on hard problems by conditioning on very off-policy prefixes. *arXiv preprint arXiv:2601.18795*, 2026.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Song, K., Tan, X., Qin, T., Lu, J., and Liu, T.-Y. Mpnet: Masked and permuted pre-training for language understanding. In *NeurIPS*, 2020.

Song, Y., Kempe, J., and Munos, R. Outcome-based exploration for llm reasoning. *arXiv preprint arXiv:2509.06941*, 2025.

Wan, Z., Shen, Y., Dou, Z., Zhou, D., Zhang, Y., Wang, X., Shen, H., Xiong, J., Tao, C., Zhong, Z., et al. Dsdr: Dual-scale diversity regularization for exploration in llm reasoning. *arXiv preprint arXiv:2602.19895*, 2026.

Wang, S., Yu, L., Gao, C., Zheng, C., Liu, S., Lu, R., Dang, K., Chen, X.-H., Yang, J., Zhang, Z., et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. In *NeurIPS*, 2025.

Wei, L., Zhang, Y., Zhang, Z., Wang, Z., Zhao, S., Huang, T., Zhao, H., Liu, C., Zhang, S., and Yan, J. Entropy-tree: Tree-based decoding with entropy-guided exploration. *arXiv preprint arXiv:2601.15296*, 2026.

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

Xing, S., Wang, S., Yang, C., Dai, X., and Ren, X. Lookahead tree-based rollouts for enhanced trajectory-level exploration in reinforcement learning with verifiable rewards. In *ICLR*, 2026.

Xu, Y. E., Savani, Y., Fang, F., and Kolter, J. Z. Not all rollouts are useful: Down-sampling rollouts in llm reinforcement learning. *arXiv preprint arXiv:2504.13818*, 2025.

Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Yang, C., Gui, L., Yang, C., Veitch, V., Zhang, L., and Zhao, Z. Let it calm: Exploratory annealed decoding for verifiable reinforcement learning. *arXiv preprint arXiv:2510.05251*, 2025.

Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, J., et al. Dapo: An open-source llm reinforcement learning system at scale. In *NeurIPS*, 2025.

Yu, S., Li, L., Zhao, W., and Yang, Z. Erpo: Token-level entropy-regulated policy optimization for large reasoning models. *arXiv preprint arXiv:2603.28204*, 2026.

Yue, Y., Chen, Z., Lu, R., Zhao, A., Wang, Z., Song, S., and Huang, G. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? In *NeurIPS*, 2025.

Zhang, Y. and Math-AI. American invitational mathematics examination (aime) 2024. https://huggingface.co/datasets/math-ai/aime24, 2024. Hugging Face dataset repository.

Zhang, Y. and Math-AI. American invitational mathematics examination (aime) 2025. https://huggingface.co/datasets/math-ai/aime25, 2025. Hugging Face dataset repository.

Zhang, Y., Yao, W., Yu, C., Liu, Y., Yin, Q., Yin, B., Yun, H., and Li, L. Improving sampling efficiency in rlvr through adaptive rollout and response reuse. *arXiv preprint arXiv:2509.25808*, 2025.

Zhao, Z., Ren, Z., Zou, J., Yang, L., Xu, Z., Ge, X., Chen, Z., Ma, X., Shi, D., Wang, S., et al. Reinforced efficient reasoning via semantically diverse exploration. *arXiv preprint arXiv:2601.05053*, 2026.

Zheng, H., Zhou, Y., Bartoldson, B. R., Kailkhura, B., Lai, F., Zhao, J., and Chen, B. Act only when it pays: Efficient reinforcement learning for llm reasoning via selective rollouts. In *NeurIPS*, 2025.

Zhu, X., Xia, M., Wei, Z., Chen, W.-L., Chen, D., and Meng, Y. The surprising effectiveness of negative reinforcement in llm reasoning. In *NeurIPS*, 2025.

Zhuang, H., Zhou, Y., Guo, T., Huang, Y., Liu, F., Song, K., and Zhang, X. Exploring multi-temperature strategies for token-and rollout-level control in rlvr. *arXiv preprint arXiv:2510.08892*, 2025.

A. Experimental Details

A.1. RL Training and Evaluation Details

We evaluate REFT as a drop-in rollout-sampling modification on top of DAPO (Yu et al., 2025) and GRPO (Shao et al., 2024). The base models are Qwen2.5-0.5B-Instruct, Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct (Yang et al., 2024), and Llama3.2-3B-Instruct (Grattafiori et al., 2024).

Training datasets. We use three math-reasoning training sets. GSM8K (Cobbe et al., 2021) contains 7,500 elementary-grade math problems and is used with rollout group size $G = 8$ for all four models. For the 7B-scale model, we train on the BigMath (Albalak et al., 2025) levels 3–5 subset, containing 121,503 problems. For the 3B-scale models, Qwen2.5-3B-Instruct and Llama3.2-3B-Instruct, we construct BigMath-Easy by restricting BigMath to levels 1–3, yielding 127,945 problems. This follows prior evidence that removing the hardest problems can provide a more effective training signal for smaller models (Albalak et al., 2025). Both BigMath subsets use rollout group size $G = 16$.

Reward. We use a verifier-based reward composed of an answer-accuracy term and a format term. For GSM8K, the accuracy reward is weighted by 2.0 and the format reward by 0.2. For BigMath and BigMath-Easy, the accuracy reward is weighted by 1.0 and the format reward by 0.2. The same reward function and weights are used for the corresponding baseline and REFT runs.

REFT configuration. Unless otherwise stated, REFT uses candidate width $N = 20$ and selected-token count $K = 4$. Thus, REFT samples four first tokens from the policy's top-20 valid first-token candidates and allocates rollout slots evenly across them. On GSM8K, where $G = 8$, this gives two continuations per selected first token. On BigMath and BigMath-Easy, where $G = 16$, this gives four continuations per selected first token. The choice $N = 20$ follows the diagnostic result in Section 3: rollout correctness remains nearly flat across the top-20 first-token ranks even though their probabilities differ by orders of magnitude. The choice $K = 4$ provides coverage of multiple continuation regions while still assigning replicated continuations to each selected first token, reducing sensitivity to a single unlucky rollout.

Decoding, optimization, and checkpoint selection. Maximum completion length is 4,096 tokens for GSM8K and 8,192 tokens for BigMath and BigMath-Easy. All rollout sampling uses temperature 1.0 and top-$p = 1.0$ (Hu et al., 2025; Huang et al., 2026; Gu et al., 2026; Fang et al., 2026). Training uses LoRA (Hu et al., 2022) with rank 32 and DeepSpeed ZeRO-3 (Rajbhandari et al., 2020) through Hugging Face Accelerate for memory efficiency. We train up to 1,000 steps and report within this training window.

Evaluation datasets and inference. GSM8K-trained models are evaluated on the full GSM8K test set (Cobbe et al., 2021). BigMath and BigMath-Easy trained models are evaluated on the full evaluation sets of MATH-500 (Lightman et al., 2024), AIME 2024 (Zhang & Math-AI, 2024), AIME 2025 (Zhang & Math-AI, 2025), and AMC 2023 (Li et al., 2024). Inference uses vLLM (Kwon et al., 2023) with temperature 0.6 and top-$p = 0.95$.

Hardware. We train Qwen2.5-7B-Instruct on 4× NVIDIA B200 GPUs. All remaining models are trained on 8× NVIDIA L40S GPUs. For consistency, all runs for the same base model use the same hardware configuration.

Table 5. Hyperparameters for RLVR training.

Hyperparameter	Value
Optimizer	AdamW-8bit
Training batch size	128
Policy learning rate	5e−6
LoRA rank	32
Rollouts per prompt	8 for GSM8K, 16 for BigMath
Max response length	4096 for GSM8K, 8192 for BigMath
Rollout temperature	1.0
Rollout top-p	1.0
Clip range $\epsilon_{\text{low}}, \epsilon_{\text{high}}$	0.2, 0.28
Policy updates per rollout	4 for GRPO, 1 for DAPO
Reward weights	GSM8K: Accuracy 2.0, Format 0.2 BigMath: Accuracy 1.0, Format 0.2
Training step	1000 steps

A.2. Runtime Overhead

Runtime efficiency. REFT does not introduce additional model calls or increase the rollout budget. It only changes how the first token is allocated within the same rollout group. As shown in Table 8, this makes the method essentially cost-free in practice, and the measured GPU-hours are slightly lower than those of standard sampling. The reason is that REFT often reaches higher accuracy (Table 1, Table 2) with shorter average completions, reducing generation length enough to offset the small overhead of first-token routing.

Table 6. Summary of models used in this paper. We report the source and license of the base models used for RLVR training and the sentence encoder used for semantic-diversity analysis.

Model	Source	License
Qwen2.5-0.5B-Instruct	Qwen (Yang et al., 2024)	Apache License 2.0
Qwen2.5-3B-Instruct	Qwen (Yang et al., 2024)	Qwen Research License
Qwen2.5-7B-Instruct	Qwen (Yang et al., 2024)	Apache License 2.0
Llama3.2-3B-Instruct	Meta (Grattafiori et al., 2024)	Llama 3.2 Community License
all-mpnet-base-v2	Sentence Transformers (Reimers & Gurevych, 2019; Song et al., 2020)	Apache License 2.0

Table 7. Summary of datasets used in this paper. We report the source and license metadata for both training and evaluation datasets.

Dataset	Source	License
GSM8K	OpenAI (Cobbe et al., 2021)	MIT
BigMath	SynthLabsAI Big-Math-RL-Verified (Albalak et al., 2025)	Apache License 2.0
MATH-500	OpenAI PRM800K / HuggingFaceH4 (Lightman et al., 2024)	MIT (source repo)
AIME 2024	math-ai (Zhang & Math-AI, 2024)	Apache License 2.0
AIME 2025	math-ai (Zhang & Math-AI, 2025)	Apache License 2.0
AMC 2023	NuminaMath eval suite (Li et al., 2024)	Apache License 2.0

Table 8. Runtime and completion-length changes of REFT. Negative relative differences indicate that REFT is lower than standard sampling. Mean relative differences are computed from the mean values in each column.

	Runtime	Completion Length		
Configuration	GPU-hour Rel. Δ	Sampling Avg.	REFT Avg.	Rel. Δ
GSM8K, Qwen2.5-3B-Instruct, DAPO	-0.02%	245.55	235.87	-3.94%
BigMath-Easy, Qwen2.5-3B-Instruct, GRPO	-0.80%	366.25	342.33	-6.53%
BigMath-Easy, Qwen2.5-3B-Instruct, DAPO	-0.42%	353.18	333.26	-5.64%
Mean	$\mathbf{-0.41\%}$	**321.66**	**303.82**	$\mathbf{-5.55\%}$

B. Rollout Construction and Objective

In standard rollout sampling, the first token after <think> is drawn from the behavior policy,

$$f \sim \pi_{\theta_{\text{old}}}(\cdot \mid x, \texttt{<think>}).$$

In REFT, this token is instead selected by the stratified sampler from the policy's own top-N valid candidates. The rollout is therefore off-policy at this single position. After the first token is fixed, the remaining continuation is sampled from the same behavior policy conditioned on that token,

$$z \sim \pi_{\theta_{\text{old}}}(\cdot \mid x, \texttt{<think>}, f).$$

Thus, the sampling mismatch is localized to the first generated token, while the continuation decoder remains unchanged.

We apply the standard GRPO/DAPO objective to the completed rollouts without an additional importance correction for the first-token sampling distribution. This is a deliberate design choice for isolating the effect of rollout construction. The data-collection sampler changes, while the downstream RL objective is kept fixed. We do not claim that this gives an unbiased estimator of the objective under standard on-policy first-token sampling. Rather, REFT replaces a faithful draw from the sharply peaked first-token prior with a controlled allocation of the same finite rollout budget across several plausible first tokens.

PPO-style clipping remains part of the underlying GRPO/DAPO objective and helps stabilize current-to-old policy updates. It should not be interpreted as an importance correction for the mismatch between the REFT first-token sampler and the original behavior-policy first-token distribution. An explicit correction toward the original first-token prior would reweight samples back toward the distribution that REFT is designed to diversify, thereby counteracting the intended budget allocation.

This setup is related in spirit to other RLVR interventions that modify rollout collection or conditioning while largely keeping the downstream optimization procedure fixed, such as temperature-scheduled rollout generation (Zhuang et al., 2025; Yang et al., 2025) and prefix-conditioned or prefix-reuse methods (Setlur et al., 2026; Huang & Wan, 2026).

In our experiments, the answer verifier, reward function, advantage computation, RL objective, model architecture, continuation decoder, and total rollout budget are otherwise unchanged. Section 5 shows that this localized sampling mismatch trains stably and improves Pass@1, Pass@8, and Pass@64 across the evaluated settings.

C. Larger-Budget Coverage

Table 9 reports the full per-benchmark Pass@64 results corresponding to the compact summary in Table 3. Pass@64 measures larger-budget coverage, namely whether at least

Table 9. Pass@64 reasoning performance across models and RLVR methods. Pass@64 denotes whether at least one of 64 sampled completions is correct. Math Avg. excludes GSM8K.

Method	GSM8K Pass@64	MATH-500 Pass@64	AIME24 Pass@64	AIME25 Pass@64	AMC23 Pass@64	Math Avg. Pass@64
Qwen2.5-3B-Instruct	98.18	87.80	33.33	36.67	90.00	61.95
GRPO	98.64	88.20	36.67	40.00	92.50	64.34
GRPO + REFT	**99.39**	**89.60**	**40.00**	**46.67**	**95.00**	**67.82**
DAPO	97.65	88.00	36.67	43.33	92.50	65.13
DAPO + REFT	**98.18**	**89.20**	**43.33**	**46.67**	**97.50**	**69.18**
Llama3.2-3B-Instruct	97.42	80.60	30.00	13.33	82.50	51.61
GRPO	97.95	81.80	33.33	20.00	85.00	55.03
GRPO + REFT	**98.56**	81.80	**36.67**	**30.00**	**87.50**	**58.99**
DAPO	97.35	82.60	36.67	23.33	82.50	56.28
DAPO + REFT	**98.48**	**83.20**	36.67	**30.00**	**87.50**	**59.34**
Qwen2.5-7B-Instruct	97.95	89.20	40.00	43.33	92.50	66.26
GRPO	98.33	90.80	43.33	46.67	95.00	68.95
GRPO + REFT	**98.94**	**92.00**	**50.00**	46.67	**97.50**	**71.54**
DAPO	**98.48**	90.60	46.67	46.67	95.00	69.74
DAPO + REFT	98.41	**91.40**	46.67	**50.00**	**97.50**	**71.39**

one of 64 sampled completions is correct. Math Avg. averages MATH-500, AIME24, AIME25, and AMC23, excluding GSM8K. Table 9 shows that the larger-budget gains are not driven by a single benchmark, and that REFT generally preserves or improves the ability to recover correct trajectories under a larger sampling budget.

D. Semantic-Diversity Computation

To test whether first-token choice affects the continuation rather than only the first word, we measure cosine semantic diversity after removing the first token itself. Each stripped continuation is embedded with `all-mpnet-base-v2` (Song et al., 2020; Reimers & Gurevych, 2019), following prior works (Kirk et al., 2024; Karouzos et al., 2026; Zhao et al., 2026; Bai et al., 2026) that compute semantic diversity using mean pairwise cosine distance between sentence embeddings.

For a rollout group $\mathcal{G}_x = \{y_i\}_{i=1}^{G}$, let $e(\text{strip}(y_i))$ denote the embedding of the continuation. We define within-prompt semantic diversity as the average pairwise cosine distance:

$$D_{\text{sem}}(\mathcal{G}_x) = \frac{2}{G(G-1)} \sum_{1 \le i < j \le G} d_{\text{sem}}(y_i, y_j), \quad d_{\text{sem}}(y_i, y_j) = 1 - \cos(e(\text{strip}(y_i)), e(\text{strip}(y_j))) . \tag{3}$$

We report the average of $D_{\text{sem}}(\mathcal{G}_x)$ over prompts. Higher values indicate more semantically diverse continuations within the rollout group.

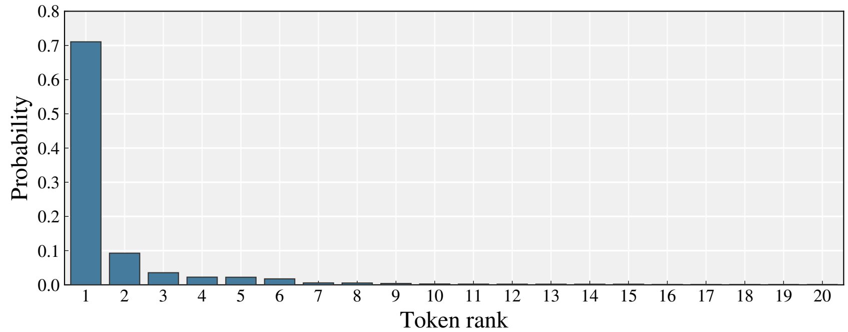
E. Additional First-Token Distributions

E.1. First-token concentration across models and datasets.

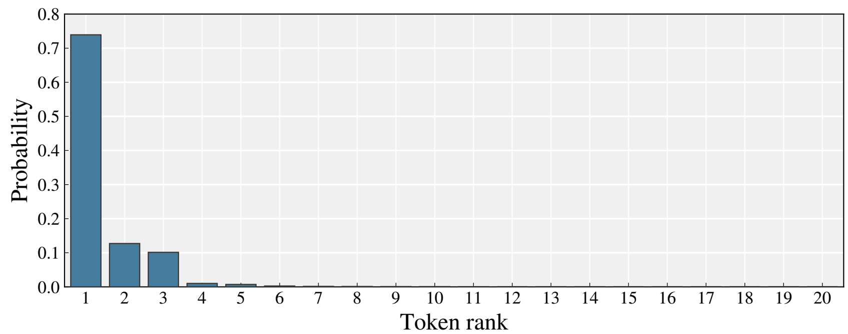
Figure 8 shows first-token rank probabilities for additional model–dataset pairs. Across Qwen2.5-3B on BigMath-Easy, Llama3.2-3B on GSM8K, and Qwen2.5-0.5B on GSM8K, the probability mass remains concentrated in the top few first-token ranks. This supports the diagnosis in Section 3 that first-token concentration is not specific to the single setting used in Figure 1.

E.2. Per-prompt First-Token Distributions

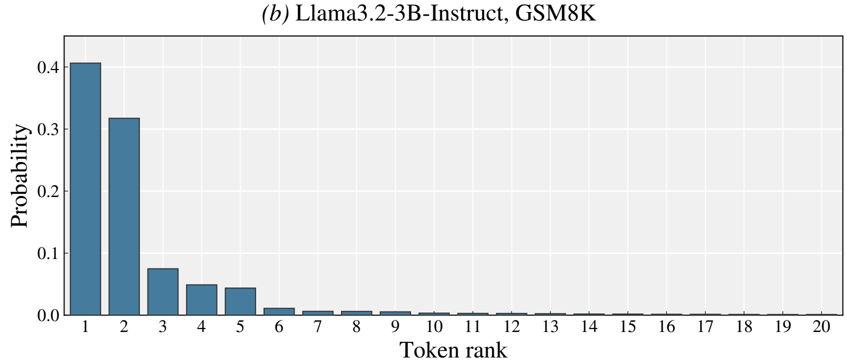
The aggregate first-token distribution in Section 3 averages over many prompts. This appendix shows that the same concentration phenomenon appears at the individual-prompt level. For each prompt, we plot the policy's probability over the top-20 first-token candidates after the reasoning marker <think>. The horizontal axis lists the actual tokens in decreasing-probability order rather than abstract ranks, making the identity of the dominant openers visible. On three GSM8K prompts under Llama3.2-3B-Instruct, almost all probability mass falls on the top few first tokens, with the remaining top-20 tokens carrying negligible probability.



(a) Qwen2.5-3B-Instruct, BigMath-Easy



(b) Llama3.2-3B-Instruct, GSM8K



(c) Qwen2.5-0.5B-Instruct, GSM8K

Figure 8. **First-token distributions across additional model–dataset pairs.** Mean first-token probability by token rank for (a) Qwen2.5-3B-Instruct on BigMath-Easy, (b) Llama3.2-3B-Instruct on GSM8K, and (c) Qwen2.5-0.5B-Instruct on GSM8K. Across settings, the first-token distribution is concentrated in the top few ranks, supporting the generality of the first-token prior concentration observed in Figure 1.

GSM8K prompt 0 **(gold answer:** 18)

Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers' market daily for $2 per fresh duck egg. How much in dollars does she make every day at the farmers' market?

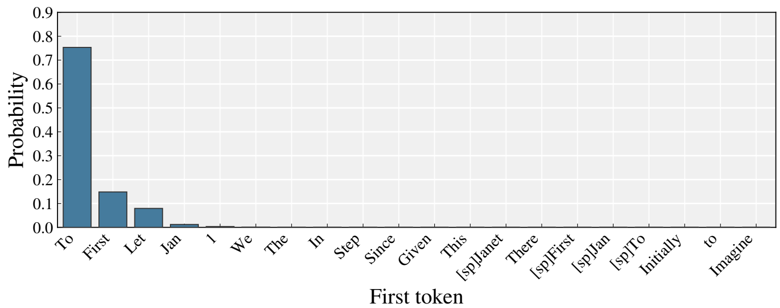


Figure 9. **Top-20 first-token probabilities for GSM8K prompt 0**, under Llama3.2-3B-Instruct.

GSM8K prompt 1 **(gold answer:** 3)

A robe takes 2 bolts of blue fiber and half that much white fiber. How many bolts in total does it take?

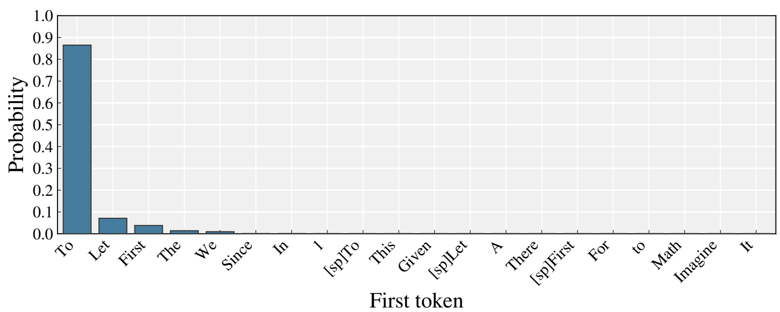


Figure 10. **Top-20 first-token probabilities for GSM8K prompt 1**, under Llama3.2-3B-Instruct.

GSM8K prompt 2 **(gold answer:** 70,000)

Josh decides to try flipping a house. He buys a house for $80,000 and then puts in $50,000 in repairs. This increased the value of the house by 150%. How much profit did he make?

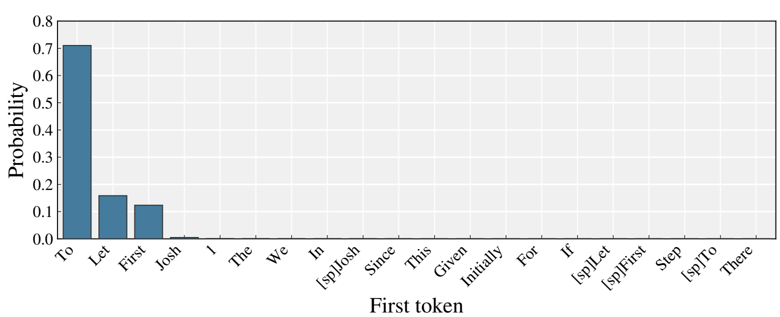


Figure 11. **Top-20 first-token probabilities for GSM8K prompt 2**, under Llama3.2-3B-Instruct.

F. Qualitative rollout examples

We present two actual GSM8K rollout examples illustrating the rollout-level effect of REFT. Both examples are sampled from Qwen2.5-3B-Instruct trained with GRPO on GSM8K. For each prompt, we contrast eight standard-sampling rollouts with eight REFT rollouts produced as $K=4$ first-token branches with $G/K=2$ continuations per branch. We display representative continuations from each condition rather than all sixteen completions.

F.1. Case 1: Water-consumption problem (gold answer = 26)

Prompt

John likes to have a glass of water with breakfast, lunch and dinner. Finally, he has one before he goes to bed as well. John does this every weekday, but on the weekends he likes to relax and have a soda with dinner instead. How many glasses of water does John drink in a week?

Standard sampling: $0/8$ **correct.** All eight rollouts begin with the same opener (`To`) or a near variant (`First`, `To solve`), and they converge on a small set of incorrect answers $(7, 15, 17, 17, 20, 22, 22, 22)$. None counts the four daily water occasions (breakfast, lunch, dinner, and before bed) correctly.

Standard rollout 1 answer

<think>
To determine the total number of glasses of water John drinks in a week, we need to break down his water consumption pattern and calculate accordingly.
John drinks one glass of water with each meal throughout the week: - For breakfast - For lunch - For dinner
[. . .]
On weekends (Saturday and Sunday), he drinks a soda instead of water during dinner: - Saturday: 1 glass of water - Sunday: 1 glass of water
5 (weekdays) + 2 (weekend days) = 7 glasses
</think> <answer> 7 </answer>

Standard rollout 6 answer

<think>
To determine how many glasses of water John drinks in a week, we need to break down his drinking habits by weekdays and weekends.
1. **Weekdays**: John drinks water with breakfast, lunch, dinner, and before bed. That gives 4 glasses per weekday $\times 5$ weekdays = 20 glasses.
[. . .]
2. **Weekends**: He has a soda with dinner on weekend days, but still drinks water at the other meals.
$20 + 2 = 22$ glasses
</think> <answer> 22 </answer>

REFT: $8/8$ **correct.** The four selected first tokens are `Let`, `First`, `Given`, `Initially`. Each opens a different framing of the same problem (step-by-step decomposition, weekday-first, premise-restatement, sequential narration), and all eight continuations reach the correct answer 26.

REFT rollout 1 first token

<think>
Let's break down the problem step by step.
1. John drinks water at the following times: breakfast, lunch, dinner, and before bed. 2. Therefore, John drinks water 4 times a day. 3. There are 5 weekdays and 2 weekend days. 4. On weekdays: $5 \times 4 = 20$ glasses. 5. On weekends, he switches from water to soda for dinner, so he drinks 3 glasses each weekend day, $2 \times 3 = 6$ glasses.
$20 + 6 = 26$ glasses.
</think> <answer> 26 </answer>

REFT rollout 5 first token

<think>
Given the information, John drinks water with his meals every weekday and a soda with his dinner on weekends. A week consists of 5 weekdays and 2 weekend days.
[. . .]
Weekdays contribute $5 \times 4 = 20$ glasses, and weekend days contribute $2 \times 3 = 6$ glasses, for a total of 26 glasses.
</think> <answer> 26 </answer>

REFT rollout 7 first token

<think>
Initially, let's determine how many glasses of water John drinks on a regular weekday. According to the problem, John drinks one glass at breakfast, lunch, dinner, and before bed, giving 4 glasses per weekday.
[. . .]
For weekends, dinner is replaced by soda, leaving 3 glasses of water per day. Total: $5 \times 4 + 2 \times 3 = 26$ glasses.
</think> <answer> 26 </answer>

Why this case is informative. Standard sampling does not fail by collapsing onto a single wrong answer. It produces several distinct wrong answers, but it never opens the trace with a framing that explicitly enumerates the four daily water occasions. REFT substitutes three of the eight "`To`"-style openings with structurally different first tokens, and each of these openings unfolds a continuation that does include the missing fourth glass. The first-token swap is what surfaces the correct decomposition; everything after the opener is sampled with the same decoder under both conditions.

F.2. Case 2: Lemon-tree problem (gold answer = 13)

Prompt

Carlos is planting a lemon tree. The tree will cost $90 to plant. Each year it will grow 7 lemons, which he can sell for $1.5 each. It costs $3 a year to water and feed the tree. How many years will it take before he starts earning money on the lemon tree?

Standard sampling: $0/8$ **correct.** Five of the eight rollouts return the same wrong answer of 12 (computing the year in which cumulative profit is non-negative, rather than the year in which it is strictly positive); the remaining three return 9, 73.5, and 1. The continuations differ in surface text but settle on the same off-by-one error.

Standard rollout 1 answer

<think>
To determine how many years it will take before Carlos starts earning money on his lemon tree, we need to set up an inequality based on his profit.
[. . .]
$7.5y \geq 90 \;\Rightarrow\; y \geq 12$
So Carlos will start earning money in year 12.
</think> <answer> 12 </answer>

REFT: $7/8$ **correct.** The selected first tokens are `First`, `Carlos`, `To`, `When`. Seven of the eight rollouts correctly compute that the tree starts *strictly earning* money in year 13.

REFT rollout 1 first token

<think>
First, we need to determine if the lemon tree will generate more money through sales than it costs to maintain.
[. . .]
Annual net profit: $7 \times 1.5 - 3 = 7.5$. We need $7.5y > 90$, i.e., $y > 12$. The smallest such integer is $y = 13$.
</think> <answer> 13 </answer>

REFT rollout 3 first token

<think>
Carlos starts by investing $90 to plant the lemon tree. Each year, he has to spend $3 to water and feed it, while earning $7 \times \$1.5 = \$10.5$ from selling lemons.
[. . .]
Cumulative net profit after y years is $7.5y - 90$, which becomes positive at $y = 13$.
</think> <answer> 13 </answer>

REFT rollout 7 first token

<think>
When calculating the number of years it will take before Carlos starts earning money on the lemon tree, we need to consider both the initial cost and the recurring annual costs and income.
[. . .]
The break-even is at $y = 12$, so Carlos first *earns* money in year 13.
</think> <answer> 13 </answer>

Takeaway. Across both cases, the difference between standard sampling and REFT is not that REFT produces stranger or longer text. The completions remain coherent English with valid `<think>`/`<answer>` structure and no degenerate behavior. The difference is that diversifying the first token surfaces alternative framings of the same problem, which in turn change which arithmetic the rollout actually performs. This is the rollout-level mechanism that the aggregate analyses in Section 6 measure quantitatively.