

Reinforce-Ada: An Adaptive Sampling Framework under Non-linear RL Objectives

Wei Xiong^{*◊} Chenlu Ye^{*◊} Baohao Liao^{*■} Hanze Dong^{*◊}

Xinxing Xu[◊] Christof Monz[■] Jiang Bian[◊] Nan Jiang[◊] Tong Zhang[◊]

[◊]University of Illinois Urbana-Champaign [◊]Microsoft Research [■]University of Amsterdam

Abstract

Reinforcement learning for large language model reasoning often suffers from signal collapse: under uniform small-group sampling, difficult prompts frequently produce uninformative all-fail or all-success groups, causing useful gradients to disappear. We show that this failure is largely a statistical artifact of undersampling rather than an inherent limitation of the model. This suggests that sampling budget should be allocated adaptively across prompts according to their pass rates. We formalize this idea through a family of non-linear RL objectives, which induce difficulty-aware weighting and recover prior log-objective analyses as a special case. Based on this view, we propose REINFORCE-ADA, with an estimation-based variant and a model-free sequential variant that adaptively allocate rollouts to recover informative signals. Across multiple models and math reasoning benchmarks, REINFORCE-ADA consistently improves over standard GRPO baselines, recovering much of the benefit of large-group training at substantially lower cost.

1 Introduction: Signal Collapse in RL with Group Sampling

We study signal collapse in reinforcement learning algorithms for large language models (LLMs). Formally, for a prompt $x \sim d_0$, the policy π_θ produces a response $a \sim \pi_\theta(\cdot|x)$, and a verifier yields a binary reward $r^*(x, a) \in \{0, 1\}$. The learning objective is

$$J(\theta) = \mathbb{E}_{x \sim d_0} [\mathbb{E}_{a \sim \pi_\theta(\cdot|x)} r^*(x, a)] =: \mathbb{E}_{x \sim d_0} p_\theta(x), \quad (1)$$

where $p_\theta(x)$ is often referred to as the pass rate and serves as a metric of prompt difficulty for the model π_θ .

Estimating its gradient requires sampling multiple responses. This group-sampling paradigm underlies many widely used LLM post-training methods, including RAFT (Dong et al., 2023), GRPO (Shao et al., 2024), and online DPO (Xiong et al., 2023b). With only a few samples per prompt, inference is affordable but the gradient is noisy; with many samples, the signal is clear but inference becomes prohibitively expensive. The trade-off between signal quality and cost is a central challenge in RL for LLMs.

To make this trade-off concrete, we focus on Group Relative Policy Optimization (GRPO) (Shao et al., 2024) to illustrate our motivation. Given a prompt x and a group of n sampled responses $\{a_i, r_i\}_{i=1}^n$, GRPO uses the per-sample gradient estimator

$$g_\theta(x, a_i) = A_{\text{GRPO}}(x, a_i) \cdot \nabla_\theta \log \pi_\theta(a_i|x), \quad (2)$$

where the group-normalized advantage is

$$A_{\text{GRPO}}(x, a_i) = \frac{r_i - \bar{r}}{\sigma_r + \varepsilon}, \quad (3)$$

^{*}Equal contribution. A detailed attribution of authorship credits is provided in Appendix A. Correspondence to Hanze Dong (hanzedong@microsoft.com) and Wei Xiong (wx13@illinois.edu).

where $\bar{r}$ and σ_r are the mean and standard deviation of the group rewards. Despite these benefits, GRPO fundamentally relies on a small and fixed n per prompt, creating a vulnerability to signal collapse. When all n samples for a prompt yield identical rewards, either all correct or all incorrect, the group mean $\bar{r}$ equals each reward r_i . Consequently, the computed advantages vanish ($A_{\text{GRPO}} = 0$), resulting in a zero gradient. Such uniform-reward scenarios arise frequently: during early training phases when the model fails on all attempts for challenging prompts, and later when it consistently succeeds on trivial ones. For instance, with a group size of 8, if the pass rate $p = 0.1$, the probability that all 8 samples are 0 is $0.9^8 \approx 43\%$. Empirically, even with $n = 32$, more than half of prompts fall into this “zero-gradient” regime as models improve (Yu et al., 2025).

Crucially, this collapse is not due to the prompts being inherently trivial or impossible; it is instead a **statistical artifact** of undersampling. With finitely many samples from $\pi_\theta(\cdot|x)$, the estimate $\hat{p}_\theta(x)$ can easily equal 0 or 1 even when the true pass rate satisfies $0 < p_\theta(x) < 1$. Training prompts are typically filtered to ensure moderate difficulty (e.g., Yang et al. (2024) retain prompts where 2–5 out of 8 responses are correct), meaning most prompts have non-trivial success probabilities. With small n , however, random fluctuations make it likely to observe all-correct or all-incorrect groups, thereby masking the true learning signal. Larger n reliably recovers these signals, but at unsustainable inference cost (Figure 1).

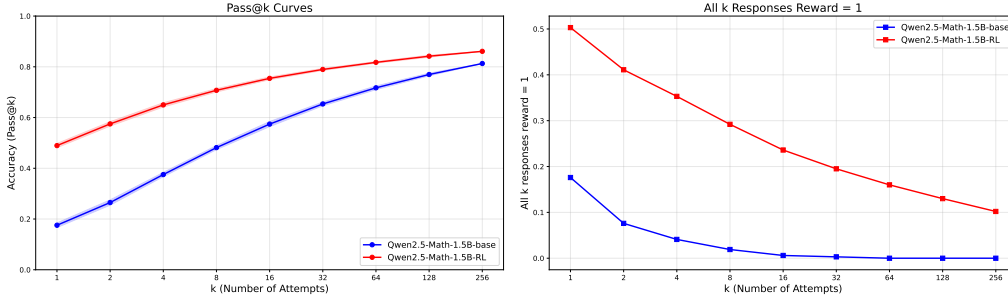


Figure 1: Pass@k curves (left) and the fraction of prompts with uniform-reward groups (right) on a subset of the Open-R1 prompt set. We compare the Qwen2.5-Math-1.5B base model with an RL checkpoint. As k increases, pass@k rises while uniform-reward groups become substantially less frequent.

These observations reveal a deeper limitation of current practice. Most existing methods use the same group size for every prompt, regardless of difficulty. This uniform allocation is simple, but statistically wasteful: easy prompts are often oversampled, while hard prompts are exactly the cases where a small group is least likely to uncover a useful learning signal. From a learning dynamics perspective, however, hard prompts are often the most informative, because progress on them changes the model where it currently fails. To balance signal recovery against computational cost, we should allocate more budget to prompts with low pass rates instead of sampling every prompt equally.

Motivated by Yao et al. (2025), who derived a $1/p$ -weighted gradient for RAFT from a statistical EM interpretation, we take a more general objective-level view that applies to general policy gradient algorithms. A convenient way to express this intuition is through a family of utility-transformed pass-rate objectives:

$$J_f(\theta) = \mathbb{E}_x[f(p_\theta(x))], \quad (4)$$

where f is a non-linear utility function of the prompt pass rate. This formulation makes the induced prompt-dependent weighting explicit in the gradient,

$$\nabla J_f(\theta) = \mathbb{E}_x[f'(p_\theta(x)) \cdot \nabla p_\theta(x)], \quad (5)$$

thereby assigning different emphasis to different pass-rate regimes. For example, the log utility $f(p) = \log p$ yields a $1/p$ weight. In this sense, the RAFT result of Yao et al. (2025) can be viewed as the log-utility special case of a broader framework. Here, f serves as an

analytical lens for expressing how strongly we wish to emphasize different prompt regimes and for deriving adaptive allocation rules from that emphasis.

Building on this framework, we propose REINFORCE-ADA, a family of adaptive sampling algorithms that realizes this weighting principle through inference budget allocation. Our contributions are threefold:

1. **A utility-based framework for prompt-dependent weighting.** We introduce a family of utility-transformed pass-rate objectives as a conceptual framework for adaptive allocation. This view makes explicit how different choices of f emphasize different prompt regimes.
2. **A broad family of variance-reduction estimators and allocation rules.** Within our framework, different choices of f induce different prompt-dependent weightings, which in turn give rise to a broad family of variance-reduction policy gradient estimators and adaptive allocation rules for general policy gradient algorithms beyond RAFT. In this sense, our framework generalizes the result of Yao et al. (2025), with the RAFT $1/p$ -weighted gradient appearing as the log-utility special case. This perspective motivates our estimation-based method REINFORCE-ADA-EST.
3. **A model-free method via sequential sampling.** We introduce REINFORCE-ADA-SEQ, in which the sampling budget is encoded implicitly in the expected stopping time of the sequential sampling process. By continuing to sample until a stopping condition is met, the method recovers signal without explicit pass-rate estimation. For example, the stopping rule “sample until one positive response is observed” yields an expected allocation proportional to $1/p$, exactly matching the log objective. Empirically, compared with standard GRPO with uniform sampling, this leads to improved compute efficiency: under the same compute budget, our method achieves better performance.

2 The Weighted Objective Framework

We now make the weighting principle from the introduction precise. Recall the general non-linear objective in Equation (4), where $f : (0, 1] \rightarrow \mathbb{R}$ is a non-decreasing function. When $f(p) = p$, we recover the standard objective. Choosing a concave f instead assigns higher marginal utility to improvements on prompts with low pass rates. Expanding the gradient in Equation (5) reveals the corresponding prompt-dependent weight:

$$\nabla J_f(\theta) = \mathbb{E}_{x \sim d_0} \left[\underbrace{f'(p_\theta(x))}_{\text{Weight } w(p)} \cdot \mathbb{E}_{a \sim \pi_\theta(\cdot|x)} \left[\nabla_\theta \log \pi_\theta(a|x) \cdot r(x, a) \right] \right]. \quad (6)$$

Different choices of f induce different weighting rules. A canonical choice is the **log objective**, $f(p) = \log p$, which gives

$$\nabla J_{\log}(\theta) = \mathbb{E}_{x \sim d_0} \left[\frac{1}{p_\theta(x)} \cdot \mathbb{E}_{a \sim \pi_\theta(\cdot|x)} \left[r^*(x, a) \nabla_\theta \log \pi_\theta(a|x) \right] \right].$$

Thus the weight is $1/p_\theta(x)$, which explicitly emphasizes difficult prompts. This is closely related to the weighted gradient derived for RAFT through the EM interpretation in Yao et al. (2025), but here it arises from a general policy-gradient objective beyond RAFT. A broader family is given by the power objective $f(p) = p^\alpha$ with $\alpha > 0$, which yields $\nabla J_\alpha(\theta) = \mathbb{E}_{x \sim d_0} \left[\alpha p_\theta(x)^{\alpha-1} \cdot \mathbb{E}_{a \sim \pi_\theta(\cdot|x)} \left[r^*(x, a) \nabla_\theta \log \pi_\theta(a|x) \right] \right]$, providing a softer form of difficulty-aware weighting.

Figure 2 visualizes representative choices of f and f' . The standard objective $f(p) = p$ gives the uniform weight $w(p) = 1$, whereas concave choices such as $\log p$ and $\sqrt{p}$ place larger weights on lower-pass-rate prompts. This is the core mechanism behind adaptive allocation in our framework.

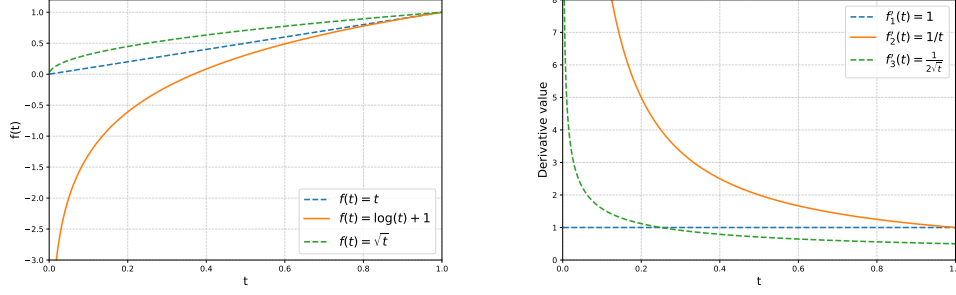


Figure 2: Visualization of different $f(t)$ and $f'(t)$. The concave functions $\log(t)$ and $\sqrt{t}$ assign larger weights $f'(t)$ to difficult prompts ($t \rightarrow 0$).

2.1 REINFORCE-ADA-EST: Estimation-based Allocation

We begin with the most direct implementation route. Suppose that the prompt pass rates p_i are already known, or have been estimated accurately enough to be treated as known for allocation. We can then use this information to adapt the sampling budget across prompts. The first application of the proposed non-linear framework is to generalize the variance-reduction view of Yao et al. (2025) from RAFT to general policy gradients and beyond the log objective. For a chosen utility f , let $\hat{g}_{\text{batch}}$ be an unbiased estimator of the weighted batch gradient in Equation (6), where prompt x_i receives n_i samples and weight $w_i = f'(p_i)$. Let $\sigma_g^2(x_i)$ denote the variance of the corresponding prompt-level gradient estimator under unit weight, namely with $w_i = 1$. Then the batch variance takes the generic form

$$V = \sum_{i=1}^B \frac{\sigma_g^2(x_i)}{n_i} w_i^2, \quad n_i^* \propto w_i \sigma_g(x_i), \quad (7)$$

where the second relation follows from minimizing V under a fixed total budget $\sum_i n_i = N$. Under the standard assumption that gradient norms are similar across prompts, $\sigma_g^2(x_i)$ is driven mainly by reward variance: $\sigma_g^2(x_i) \propto p_i$ for RAFT and $\sigma_g^2(x_i) \propto p_i(1 - p_i)$ for policy gradient with a baseline, where the baseline further reduces variance through centering. For the log utility, $w_i = 1/p_i$, which recovers the RAFT rule of Yao et al. (2025) and extends it to policy gradient:

$$n_i^* \propto \sqrt{\frac{1}{p_i}} \quad (\text{RAFT}), \quad n_i^* \propto \sqrt{\frac{1-p_i}{p_i}} \quad (\text{policy gradient}).$$

For the power utility $f(p) = p^\alpha$, the same argument gives $n_i^* \propto \alpha p_i^{\alpha-1} \sigma_g(x_i)$. In contrast, the GRPO normalization corresponds to an implicit weight proportional to $1/\sqrt{p_i(1-p_i)}$, which can be associated with the utility $2 \arcsin(\sqrt{p_i})$. Plugging this weight into the same framework gives $n_i^* \propto 1$, so uniform allocation is exactly the variance-optimal rule in that case. Intuitively, this happens because GRPO already normalizes by the within-group standard deviation. REINFORCE-ADA-EST implements this objective-conditioned variance-reduction idea directly: it first estimates prompt pass rates, and then allocates sampling budget according to the induced optimal rule. We defer the full derivations, additional cases, pass-rate estimation methods, and proofs to Appendix D and Appendix C.2.

However, these examples also make clear that variance-optimality is defined relative to the chosen utility. Different choices of f induce different optimal estimators and allocation rules. As shown by Yao et al. (2025), the log-weighted rule can outperform uniform GRPO in pass rate. When our primary interest is pass rate itself, this suggests that the tight coupling between utility and allocation can be risky, since the variance-optimal rule for a chosen utility need not be the one that best improves pass rate. Moreover, methods of this kind require estimating p , or equivalently the value of a prompt, in advance. This typically incurs additional cost, either through extra inference, as in Yao et al. (2025) where

part of the sample budget is used to estimate pass rates, or through training an auxiliary value network. Finally, the variance analysis above also simplifies away the dependence on prompt-level gradient norms. In principle, the optimal rule depends on these norms as well, but they are high-dimensional and unlikely to be estimated robustly from limited samples.

2.2 REINFORCE-ADA-SEQ: Implicit Allocation via Sequential Sampling

We now step back from direct objective-specific optimization and instead use the utility view more as a lens for how strongly to emphasize different pass-rate regimes. This leads to a model-free alternative that integrates estimation and adaptive allocation into a single online sampling process.

From this perspective, the utility f acts as a control knob for adaptive allocation. Rather than taking J_f as a literal optimization target, we use f to specify a desired relative emphasis across prompts, and interpret the induced weight $f'(p)$ as a target allocation pattern for inference compute. Different choices of f therefore control behavior in the pass-rate tails. For example, $f(p) = \log p$ gives $f'(p) = 1/p$ and strongly emphasizes difficult prompts, whereas $f(p) = \sqrt{p}$ yields a milder allocation rule. More generally, one-sided utilities emphasize the difficult tail near $p \rightarrow 0$, while two-sided utilities can also increase allocation near the easy tail when negative evidence becomes rare. Figure 2 can be read directly from this perspective: the curves of $f'(t)$ visualize how aggressively each utility allocates relative compute across pass-rate regimes.

To instantiate this view, we begin with the log utility, which induces the allocation rule $1/p$ and therefore places strong emphasis on difficult prompts. The most direct way to realize this rule would be to first estimate prompt pass rates and then allocate inference compute proportionally to $1/p$. A key observation, however, is that in the RAFT-style binary-success setting, the same $1/p$ allocation rule can also be realized implicitly through a simple sequential procedure: keep sampling until the first positive response is observed. The expected stopping time of this rule is $\mathbb{E}[N_i] = 1/p_i$. This gives the same allocation preference suggested by the log utility while simultaneously recovering a usable training signal.

This observation suggests a broader algorithmic principle, reminiscent of successive elimination methods in the multi-armed bandit literature (Slivkins et al., 2019). Rather than separate pass-rate estimation from allocation, we can integrate them into a single online sampling process by designing exit conditions whose expected stopping times realize the desired allocation pattern. From the signal-recovery perspective, the rule is simple: keep sampling until the prompt has produced enough evidence to support a stable update. For REINFORCE-with-baseline style updates, mixed outcomes are the natural signal unit, so a single positive sample is no longer the right stopping criterion. This motivates a two-sided exit condition that continues sampling until both positive and negative evidence have been observed. In the simplest case, the expected stopping time scales as $\frac{1}{p_i} + \frac{1}{1-p_i} - 1$, which increases allocation in both tails and closely matches the weighting induced by the two-sided utility $\log \frac{p_i}{1-p_i}$ up to an additive constant. Algorithm 1 in Appendix C gives the full procedure. At a high level, the method operates over a batch of prompts in sequential rounds:

1. **Initialization:** All prompts in the current batch begin in an *active set* (active arms).
2. **Iterative Sampling:** In each round, we generate M new responses for every prompt currently in the *active set*.
3. **Elimination:** At the end of each round, active prompts are checked against an exit condition, and those that satisfy it are removed from future rounds. We consider two variants:
 - **Positive-focused** (REINFORCE-ADA-SEQ-POS): A prompt is deactivated once we collect at least K_{pos} correct responses.

- **Balanced** (REINFORCE-ADA-SEQ-BALANCE): A prompt is deactivated once at least K_{pos} correct **and** K_{neg} incorrect responses are collected.

This process continues until all prompts are resolved or a maximum budget is reached.¹

Practical Implementation via Static Batches. A practical challenge is that sequential stopping produces random batch sizes, which are inconvenient for static GPU training. A second motivation is empirical: retaining too many negative samples in the final training batch can make optimization unstable, consistent with the comparison in Figure 5 and the discussion of Xiong et al. (2025). We therefore use all collected samples to compute a robust global baseline $\hat{p}_i = N_{\text{pos}}/N_i$, which helps recover the signal, and then systematically down-sample the response pool to a fixed update group of size n while keeping both positive and negative samples when possible.

Down-sampling introduces selection bias, since some collected responses are more likely than others to be retained in the final fixed-size update batch. We correct this with inverse-probability weighting, dividing each retained sample by its inclusion probability under the down-sampling scheme. If we also wish to preserve a utility-induced emphasis after fixing the batch size, a natural weighted estimator is

$$\hat{g}_f(x_i) = w_f(\hat{p}_i) \cdot \frac{1}{n} \sum_{j \in \mathcal{B}_i} \frac{1}{q_j} \nabla \log \pi(a_j | x_i) (r_j - \hat{p}_i).$$

Here $w_f(\hat{p}_i) = f'(\hat{p}_i)$ is the utility-induced weight, and q_j is the probability that sample j is retained by the down-sampling step. The key point, however, is that this explicit weight is optional rather than essential in REINFORCE-ADA-SEQ. The main motivation of the method is signal recovery: sequential sampling already directs more inference compute to prompts that require more evidence, regardless of whether we explicitly optimize a non-linear objective. In this sense, we use f primarily as a utility function for inference allocation rather than as the RL objective itself. When we want closer adherence to a chosen non-linear objective, we include the factor $w_f(\hat{p}_i)$. When pass rate itself is the primary concern, omitting this factor is also a natural design choice. This lets REINFORCE-ADA-SEQ retain the throughput of static batches while keeping the explicit weighting flexible.

Before turning to the large-scale LLM experiments, we also conduct detailed simulation studies to examine the sampling cost and estimation bias induced by these adaptive procedures; see Appendix E.

3 Experiment

Experimental setup. We evaluate REINFORCE-ADA on three backbones, Qwen2.5-Math-1.5B, Qwen2.5-Math-7B, and Llama-3.2-3B-Instruct, using MATH500 (Hendrycks et al., 2021), Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), and an AIME-like set of 230 recent competition problems (Balunović et al., 2025). Training uses a curated subset of Open-R1-Math-220K² with standard deduplication, verifier filtering, and difficulty-based filtering to retain challenging prompts. Unless otherwise specified, each run is trained for a fixed budget of 600 optimization steps. All evaluations report Ave@32 with temperature 1.0 and a maximum generation length of 4096. In the main comparisons, our primary scalar metric is the weighted benchmark pass@1 aggregated across the evaluation suite.

Comparison protocol. All methods share the same underlying RL recipe in ver1 (Sheng et al., 2024); we vary only the sampling mechanism. We use two complementary comparison protocols. Our main table reports a fixed-budget comparison under the default

¹We also tested a more complex intra-round estimate-and-allocate variant, but it did not yield clear gains over this simpler implementation.

²<https://huggingface.co/datasets/open-r1/OpenR1-Math-220k>

Method	Ave. Inference	Ave. Train	Ave. Cost
GRPO-4	4	4	16
ADA-SEQ-POS	7.2	4	19.2
ADA-SEQ-BALANCE	11.1	4	23.1
GRPO-8 / ADA-EST-n8	8	8	32
GRPO-16	16	16	64

Table 1: Comparison of average inference and training cost across methods. Ave. Cost is computed as Inference + $3 \times$ Train, where the factor 3 is a rough FLOPs proxy for training relative to a forward pass. For REINFORCE-ADA-SEQ, the reported values are averaged over prompts and training steps in the Qwen2.5-Math-1.5B runs. Costs are counted by the number of responses, so the estimate is approximate because response lengths vary.

configurations in Table 2: GRPO uses group sizes $n \in \{4, 8, 16\}$, ADA-SEQ-POS and ADA-SEQ-BALANCE use a maximum inference size of $N_{\max} = 32$ together with a sub-sampled training group size of $n = 4$, where ADA-SEQ-POS uses the stopping rule $K_{\text{pos}} = 1$ and ADA-SEQ-BALANCE uses the balanced stopping rule $K_{\text{pos}} = K_{\text{neg}} = 2$. In all reported ADA-SEQ experiments, we do not apply an additional prompt-dependent weight in the final advantage. The reason is empirical as well as conceptual: the appendix ablation in Section G.4 shows that adding the GRPO-style standard-deviation normalization to scaled ADA-SEQ-BALANCE hurts pass rate, suggesting that in at least some settings directly optimizing the associated non-linear objective can be less aligned with pass rate itself. For ADA-EST, by contrast, we follow the RAFT-style log- p construction of Yao et al. (2025), so the allocation rule is kept consistent with their analysis. At the same time, we simplify the pass-rate estimation procedure and do not spend a separate inference stage purely to estimate p in advance. ADA-EST-n8 therefore allocates budget from online pass-rate estimates and is most directly comparable to GRPO-n8. This protocol is intended to compare final performance under a common 600-step training budget. We refer readers to Appendix C.2 and Appendix F for the full implementation details.

To study compute scaling directly, we construct weighted benchmark pass@1-cost curves under a separate scaling protocol. GRPO uses larger group sizes $n \in \{16, 32\}$, while ADA-SEQ-BALANCE uses matched configurations with maximum inference size $2n$, sub-sampled training group size $n/2$, and stopping thresholds $K_{\text{pos}} = K_{\text{neg}} = n/4$. We report these results separately because they are designed to trace a compute frontier rather than match the default fixed-budget benchmark. The main text only summarizes the setup; full details are deferred to Appendix F.

3.1 Main Results

Fixed-budget comparison at 600 training steps. Table 2 reports the final performance of all methods under a common budget of 600 optimization steps. Because all runs are trained for the same number of steps, the average per-step cost is proportional to the total training cost. Using the Qwen2.5-Math-1.5B runs in Table 1 as a representative example, the average cost over 600 steps follows a clear compute ladder:

$$\text{GRPO-4} < \text{ADA-SEQ-POS} < \text{ADA-SEQ-BALANCE} < \text{GRPO-8/ADA-EST-n8} < \text{GRPO-16}.$$

Within the GRPO family, increasing the group size usually improves final test accuracy, which is consistent with our signal-recovery intuition: larger groups are more likely to recover informative gradients that would otherwise collapse under small-group sampling.

Adaptive allocation is more compute-efficient. Compared with uniform sampling, adaptive allocation is more compute-efficient. At approximately matched cost, ADA-EST-n8 consistently outperforms GRPO-n8, and ADA-SEQ-BALANCE is cheaper than GRPO-n8 while outperforming it on all three backbones. On Qwen2.5-Math-1.5B, ADA-SEQ-BALANCE even slightly surpasses the much more expensive GRPO-n16. These comparisons show that adaptive sampling recovers much of the benefit of scaling up group size,

Model	Algorithm	Math500	Minerva Math	Olympiad Bench	AIME-like	Weighted Average
<i>Qwen2.5-Math-1.5B</i>	GRPO-n4	74.2	34.4	38.4	16.2	45.3
	GRPO-n8	75.3	34.3	39.2	17.3	46.1
	GRPO-n16	76.9	35.1	40.4	17.9	47.3
	ADA-SEQ-POS	75.8	35.7	38.6	16.5	46.1
	ADA-SEQ-BALANCE	77.4	36.5	40.5	17.5	47.6
	ADA-EST-n8	76.1	34.8	39.1	18.5	46.6
<i>Qwen2.5-Math-7B</i>	GRPO-n4	82.5	45.0	45.2	22.3	53.0
	GRPO-n8	82.3	44.8	45.2	23.7	53.2
	ADA-SEQ-POS	82.7	45.1	46.7	23.7	54.2
	ADA-SEQ-BALANCE	84.0	45.2	47.1	23.7	54.6
	ADA-EST-n8	83.6	42.1	46.6	24.1	53.8
<i>Llama-3.2-3B-instruct</i>	GRPO-n4	52.4	21.1	20.5	5.9	28.1
	GRPO-n8	53.7	21.6	20.9	7.3	28.9
	GRPO-n16	55.2	23.2	22.3	6.8	30.1
	ADA-SEQ-POS	52.6	22.2	21.0	7.5	28.8
	ADA-SEQ-BALANCE	53.2	22.4	21.2	8.0	29.1
	ADA-EST-n8	54.1	22.2	22.1	7.5	29.7

Table 2: Performance comparison of GRPO and REINFORCE-ADA. We report average@32 accuracy with a sampling temperature of 1.0 and a maximum generation length of 4096 tokens. The weighted average score is computed according to the number of prompts in each benchmark. ADA-SEQ-POS and ADA-SEQ-BALANCE use a maximum inference size of $N_{\max} = 32$ and a sub-sampled training group size of $n = 4$; ADA-SEQ-POS uses the positive stopping rule $K_{\text{pos}} = 1$, while ADA-SEQ-BALANCE uses the balanced stopping rule $K_{\text{pos}} = K_{\text{neg}} = 2$.

but does so more efficiently than uniform sampling under a fixed 600-step training budget. They also motivate our focus on ADA-SEQ-BALANCE in the scaling study below.

3.2 Scaling ADA-SEQ-BALANCE Against GRPO

The fixed-budget comparison above also identifies ADA-SEQ-BALANCE as the strongest sequential variant to scale further. It is consistently stronger than ADA-SEQ-POS, and it also preserves policy entropy better than GRPO in the reward–entropy analysis. We therefore use ADA-SEQ-BALANCE as the representative adaptive method in the scaling study.

Following the separate scaling protocol introduced in the setup, we compare ADA-SEQ-BALANCE against larger-group GRPO baselines on Qwen2.5-Math-1.5B. We visualize two adaptive configurations, with training keep sizes 8 and 16, against GRPO- $n16$ and GRPO- $n32$, respectively. Figure 3 plots weighted benchmark pass@1 against log-compute. The main pattern is that the smaller ADA-SEQ-BALANCE configuration is close to, and sometimes slightly better than, GRPO- $n16$ in the low-to-mid compute regime; GRPO- $n16$ remains a strong baseline overall; and GRPO- $n32$ is clearly less compute-efficient. This picture is also consistent with the auxiliary entropy analyses in Appendix G, which suggest that adaptive allocation improves learning efficiency even when the gains in weighted benchmark pass@1 are moderate.

3.3 Typical Sampling Dynamics of ADA-SEQ

To understand why the sequential method works, we next examine its sampling dynamics on a single representative model, Qwen2.5-Math-1.5B. Figure 4 tracks the extra samples generated beyond the first-round GRPO budget, together with the composition of prompts that remain unresolved under the balanced stopping rule. Early in training, the additional budget is mainly spent on difficult prompts that are still missing positive evidence. Later, as many prompts become easy and negative responses turn rare, the unresolved set shifts toward prompts missing negative evidence. As a result, ADA-SEQ-BALANCE first allocates more effort to hard prompts and later returns to easy prompts to recover rare failure signals, which helps prevent all-correct collapse and maintain a useful gradient throughout training.

Additional Results and Ablations. We defer additional reward–entropy trade-off, Pass@ k , and training-difficulty ablations to Appendix G.

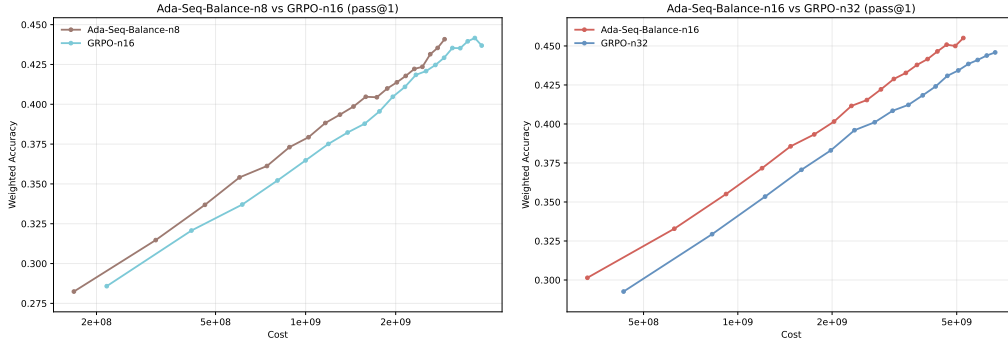


Figure 3: Weighted benchmark pass@1-compute trajectories on Qwen2.5-Math-1.5B under the scaling protocol. The horizontal axis is log-compute, measured by a token-based FLOPs proxy that combines rollout sampling cost with training update cost. The vertical axis is weighted benchmark pass@1. Left: REINFORCE-ADA with keep size 8 and maximum inference size 32 versus GRPO-n16. Right: REINFORCE-ADA with keep size 16 and maximum inference size 64 versus GRPO-n32. The main pattern is that adaptive sampling reaches a favorable efficiency regime close to GRPO-n16, while GRPO-n32 is substantially less compute-efficient and less stable.

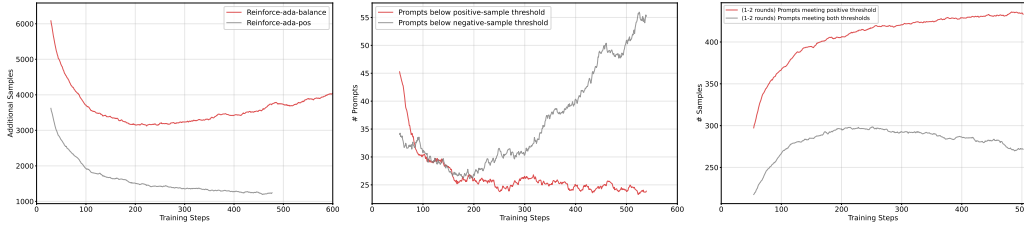


Figure 4: Typical sampling dynamics of ADA-SEQ on Qwen2.5-Math-1.5B. Left: extra samples generated in later rounds beyond the first-round GRPO budget. Middle: number of prompts that remain unresolved under ADA-SEQ-BALANCE, split by whether the stopping rule is blocked by missing positive or missing negative evidence. Right: number of prompts that can already be resolved within the first two rounds. All curves are smoothed using a moving average with a window size of 20.

Discussion and End Note. In this work, we identify signal loss under small-group sampling as a central bottleneck in RL post-training for LLMs. When the group size is small, difficult prompts often produce collapsed all-fail or all-success groups, causing the policy-gradient signal to disappear. Our main conclusion is that adaptive sampling offers an efficient way to allocate additional inference compute for signal recovery, concentrating rollouts on prompts for which informative evidence is hardest to obtain. To formalize this intuition, we introduced a family of non-linear utilities over prompt pass rate. We first used this framework to generalize objective-conditioned variance-reduction analyses beyond RAFT, and then stepped back to use it more softly as an analytical lens for designing adaptive allocation rules. In this sense, the framework not only generalizes the log-weighted insight of Yao et al. (2025) from RAFT to general policy gradients, but also provides a richer perspective on the relationship between learning objectives and compute allocation, clarifying how different objective choices induce different allocation rules and trade-offs. This broader view in turn motivates our model-free sequential method ADA-SEQ. Across multiple models and math benchmarks, REINFORCE-ADA improves compute efficiency relative to uniform group sampling, recovering much of the benefit of larger-group training without paying its full cost. Our method is local in that it allocates responses within each prompt, while many recent data-curation strategies operate at the macro, prompt level; we believe these two perspectives are complementary and should ultimately be combined as a single system.

References

- Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions, February 2025. URL <https://matharena.ai/>.
- Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. Gpg: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*, 2025.
- Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.
- Damek Davis and Benjamin Recht. What is the objective of reasoning with reinforcement learning? *arXiv preprint arXiv:2510.13651*, 2025.
- Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, KaShun SHUM, and Tong Zhang. RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL <https://openreview.net/forum?id=m7p507zb1Y>.
- Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*, 2024.
- Yunzhen Feng, Ariel Kwiatkowski, Kunhao Zheng, Julia Kempe, and Yaqi Duan. Pilaf: Optimal human preference sampling for reward modeling. *arXiv preprint arXiv:2502.04270*, 2025.
- Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 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. *arXiv preprint arXiv:2103.03874*, 2021.
- Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.
- Wenke Huang, Quan Zhang, Yiyang Fang, Jian Liang, Xuankun Rong, Huanjin Yao, Guancheng Wan, Ke Liang, Wenwen He, Mingjun Li, et al. Mapo: Mixed advantage policy optimization. *arXiv preprint arXiv:2509.18849*, 2025.
- Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL <https://github.com/huggingface/open-r1>.
- Hynek Kydlíček. Math-Verify: Math Verification Library. URL <https://github.com/huggingface/math-verify>.
- 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, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.
- Yizhi Li, Qingshui Gu, Zhoufutu Wen, Ziniu Li, Tianshun Xing, Shuyue Guo, Tianyu Zheng, Xin Zhou, Xingwei Qu, Wangchunshu Zhou, et al. Treepo: Bridging the gap of policy optimization and efficacy and inference efficiency with heuristic tree-based modeling. *arXiv preprint arXiv:2508.17445*, 2025.

- Gongrui Nan, Siye Chen, Jing Huang, Mengyu Lu, Dexun Wang, Chunmei Xie, Weiqi Xiong, Xianzhou Zeng, Qixuan Zhou, Yadong Li, et al. Ngrpo: Negative-enhanced group relative policy optimization. *arXiv preprint arXiv:2509.18851*, 2025.
- Yun Qu, Qi Wang, Yixiu Mao, Vincent Tao Hu, Björn Ommer, and Xiangyang Ji. Can prompt difficulty be online predicted for accelerating rl finetuning of reasoning models? *arXiv preprint arXiv:2507.04632*, 2025.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, YK Li, Y Wu, and Daya Guo. 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. *arXiv preprint arXiv: 2409.19256*, 2024.
- Ruizhe Shi, Runlong Zhou, and Simon S Du. The crucial role of samplers in online direct preference optimization. *arXiv preprint arXiv:2409.19605*, 2024.
- Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. Efficient reinforcement finetuning via adaptive curriculum learning. *arXiv preprint arXiv:2504.05520*, 2025.
- Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023.
- Aleksandrs Slivkins et al. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.
- Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. 2023a.
- Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. In *Forty-first International Conference on Machine Learning*, 2023b.
- Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025.
- Yixuan Even Xu, Yash Savani, Fei Fang, and Zico Kolter. Not all rollouts are useful: Down-sampling rollouts in llm reinforcement learning. *arXiv preprint arXiv:2504.13818*, 2025.
- Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun Ma, and Bo An. Simpletir: End-to-end reinforcement learning for multi-turn tool-integrated reasoning. *arXiv preprint arXiv:2509.02479*, 2025.
- An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.
- Jiarui Yao, Yifan Hao, Hanning Zhang, Hanze Dong, Wei Xiong, Nan Jiang, and Tong Zhang. Optimizing chain-of-thought reasoners via gradient variance minimization in rejection sampling and rl. *arXiv preprint arXiv:2505.02391*, 2025.
- Chenlu Ye, Zhou Yu, Ziji Zhang, Hao Chen, Narayanan Sadagopan, Jing Huang, Tong Zhang, and Anurag Beniwal. Beyond correctness: Harmonizing process and outcome rewards through rl training. *arXiv preprint arXiv:2509.03403*, 2025.

- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.
- Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.
- Chuheng Zhang, Wei Shen, Li Zhao, Xuyun Zhang, Lianyong Qi, Wanchun Dou, and Jiang Bian. Policy filtration in rlhf to fine-tune llm for code generation.
- Ruiqi Zhang, Daman Arora, Song Mei, and Andrea Zanette. Speed-rl: Faster training of reasoning models via online curriculum learning. *arXiv preprint arXiv:2506.09016*, 2025.
- Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025a.
- 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. *arXiv preprint arXiv:2506.02177*, 2025b.
- Han Zhong, Yutong Yin, Shenao Zhang, Xiaojun Xu, Yuanxin Liu, Yifei Zuo, Zhihan Liu, Boyi Liu, Sirui Zheng, Hongyi Guo, et al. Brite: Bootstrapping reinforced thinking process to enhance language model reasoning. *arXiv preprint arXiv:2501.18858*, 2025.
- Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in llm reasoning. *arXiv preprint arXiv:2506.01347*, 2025.

A Authorship and Credit Attribution

All authors provided valuable contributions to this project, each bringing unique expertise and insights that were crucial for its success.

WX proposed the project idea, initiated and organized the project, and developed the core sequential sampling algorithm; developed the framework of non-linear RL objective; implemented the initial codebase, processed the data, and ran proof-of-concept experiments to validate its effectiveness; drafted the initial version of the paper, with subsequent contributions and revisions from co-authors.

CY proposed the idea of global normalization; jointly developed the core sequential sampling algorithms, including the codes for Reinforce-Ada-Seq-pos, Reinforce-Ada-Seq-balance, global normalization, and verification; ran the experiments for Qwen2.5-Math-1.5B and Qwen2.5-Math-7B; contributed to the paper writing.

BL processed the data; ran the experiments for Qwen2.5-Math-7B, Llama, Qwen3-4B; provided the released version of code (including the Tinker version); and contributed to the paper proofreading.

HD initiated, coordinated, and drove the project; provided insights about the algorithm and project design; designed and implemented the production-ready versions of REINFORCE-ADA; developed and refactored a scalable and configurable codebase; conducted key experiments for Qwen2.5-Math-1.5B; made substantial contributions to the manuscript writing.

XX, CM, JB, NJ, TZ are senior authors, supported and advised the work, provided resources, and suggested experiments and improvements to the writing.

B Related Work

Data filtering and selection in online RL training for LLMs. Our work is related to the growing body of literature on data selection and filtering for online reinforcement learning with LLMs, which also traces back to RLHF studies (Zhang et al.; Xiong et al., 2023a; Dong et al., 2024; Shi et al., 2024; Feng et al., 2025). In the context of RLVR, some methods employ an oversample-then-downsample strategy: they first generate a large, uniform set of responses for each prompt and then select a subset based on specific criteria. Xu et al. (2025) propose downsampling to maximize reward variance within the group, while Ye et al. (2025) use process rewards to select positive samples, mitigating issues with falsely correct responses. Xue et al. (2025) study data filtering in the context of tool-integrated reasoning, where they find that trajectories with invalid tool calls significantly hurt training stability. Li et al. (2025) propose using process search to branch trajectories and collect data.

Addressing signal loss in GRPO. A central challenge in applying GRPO is the “signal loss” problem, where groups of responses with uniform rewards yield zero gradients. Prior work has identified this issue and proposed several solutions. The most direct approach is passively filtering out these prompts (Yu et al., 2025; Xiong et al., 2025). Among these methods, DAPO (Yu et al., 2025) is the closest to ours in spirit, since both methods respond to signal collapse through adaptive data collection. However, the two mechanisms operate at different levels. DAPO keeps a fixed group size for each prompt, repeatedly samples new prompts, discards prompt groups with no learning signal, and accumulates only valid groups until reaching a target batch size. In this sense, DAPO improves data efficiency after signal collapse has already occurred. In contrast, REINFORCE-ADA operates at the within-prompt allocation level: it actively allocates more responses to prompts whose signals are hard to recover, with the goal of preventing or repairing collapse rather than filtering it out afterward. The two ideas are complementary, since DAPO acts on prompt selection while our method acts on response allocation within each prompt. Another line of work proposes modifying the advantage computation to avoid zero gradients. To avoid discarding samples, Nan et al. (2025) propose augmenting the reward group with a constant (the maximum possible reward), ensuring the variance is not zero by introducing a bias. Le

et al. (2025) propose assigning advantages based on entropy information for prompts with uniform rewards. Finally, some works mitigate this issue by selectively choosing prompt batches. For instance, Qu et al. (2025) employ a Bayesian framework to predict a prompt’s pass rate and selectively sample informative prompts during online training. Our work differs by tackling the problem at the collection stage itself, ensuring that a sufficient signal is gathered adaptively rather than correcting for its absence afterward. Similarly, Shi et al. (2025) propose using adaptive curriculum learning to select prompts with suitable difficulty during online RL training. Zhang et al. (2025) also develop curriculum-learning methods to select training prompts of intermediate difficulty in a two-stage manner. Zheng et al. (2025b) use a dictionary-based approach to record historical reward from the last epoch and skip uninformative prompts.

Learning objective in RLVR. A key contribution of this work is the systematic formulation of non-linear RL objectives and the weighted gradient estimators that arise from them. Our theoretical framework builds directly on the foundational insights of Yao et al. (2025), which is one of the earliest works to recognize that RLVR can be viewed through a prompt-dependent weighting scheme from a theoretical perspective. In particular, Yao et al. (2025) interpret rejection sampling fine-tuning (RAFT) (Dong et al., 2023) under an Expectation-Maximization (EM) view (Singh et al., 2023; Zhong et al., 2025) and make the important observation that the true expected gradient in EM-RAFT is weighted by $1/p_\theta(x)$. They further characterize the variance structure under this weighted gradient and show that the optimal sampling budget should scale with $1/\sqrt{p_\theta(x)}$. Motivated by their results, our early attempts tried to extend their optimal variance-reduction estimator to the standard Reinforce algorithm, but this turned out to be infeasible. This led us to analyze the log-objective for Reinforce, which naturally recovers a variance structure similar to EM-RAFT. Our framework also captures and generalizes the result of Yao et al. (2025) (see Appendix D for details).

From an algorithmic perspective, Yao et al. (2025) propose an improved RAFT variant with variance-reduced gradient estimates. However, their method follows a two-stage “explore-then-exploit” procedure: a small portion of the budget is used to estimate pass rates (often offline), and the remaining budget is allocated based on these estimates. This design has a key limitation—when the initial budget is small, the pass-rate estimate has very high variance, especially for difficult prompts (low p_i), where the initial sample almost always contains zero positive responses. In contrast, we study the more general Reinforce algorithm, clarify the connection between explicit weighting and implicit weighted sampling, and provide two practical realizations—an estimation-based method and a model-free sequential sampling method—that mitigate this issue.

We also note a very recent theoretical note by Davis & Recht (2025), which appeared concurrently with our second preprint version. Their work also observes that non-linear objectives can be linked to heuristic weighting schemes used in algorithms such as GRPO. In comparison, we not only establish the theoretical link but also explicate the duality between optimizing this objective via explicit weights versus adaptive sampling. Furthermore, we go beyond theory to translate these insights into a practical, efficient algorithm, REINFORCE-ADA, and rigorously validate its effectiveness through extensive experiments.

GRPO variant designs. Our work is orthogonal to another line of research focusing on modifying the policy gradient algorithm itself, such as innovations in advantage estimation and clipping mechanisms (Hu, 2025; Zhu et al., 2025; Zheng et al., 2025a; Huang et al., 2025; Chu et al., 2025). While these methods refine the core update rule, we focus on the data generation and construction pipeline. Our adaptive sampling framework is complementary to these algorithmic improvements and can be combined with them.

C Additional Details on Algorithm Design

C.1 REINFORCE-ADA-SEQ

Please refer to Algorithm 1.

Algorithm 1 REINFORCE-ADA-SEQ (one training iteration)

```

1: Input: Current policy  $\pi_\theta$ , batch of prompts  $\mathcal{D}$ , effective group size for update  $n$ , number of sampling rounds  $N$ , samples per round  $M \geq n$ , and exit condition function  $\text{ExitCondition}(\cdot)$ 
2: PHASE 1: ADAPTIVE SAMPLING DATA COLLECTION
3: Set all prompts  $x$  as active and initialize response set  $\mathcal{S}_x \leftarrow \emptyset$ 
4: for  $t = 1, \dots, N$  do ▷ Iterate through sampling rounds
5:   for each prompt  $x \in \mathcal{D}$  where  $\text{active}(x)$  is true do
6:     Sample  $M$  responses  $\{a_j, r_j\}_{j=1}^M \sim \pi_\theta(\cdot | x), r^*(\cdot, \cdot)$ 
7:     Add to collection:  $\mathcal{S}_x \leftarrow \mathcal{S}_x \cup \{a_j, r_j\}_{j=1}^M$ 
8:     if  $\text{ExitCondition}(\mathcal{S}_x)$  is met then
9:       Mark prompt as inactive:  $\text{active}(x) \leftarrow \text{false}$ 
10:    end if
11:  end for
12: end for
13: PHASE 2: TRAINING BATCH AND OBJECTIVE CONSTRUCTION
14: Initialize an empty set for the final training data:  $\mathcal{B} \leftarrow \emptyset$ 
15: for each prompt  $x \in \mathcal{D}$  do ▷ Use all collected samples ("global statistics") for normalization
16:   Let  $\mathcal{S}_x = \{(a_j, r_j)\}_{j=1}^{|\mathcal{S}_x|}$  be the full set of collected samples for prompt  $x$ 
17:   Compute the global baseline:  $\hat{p}_x \leftarrow \frac{1}{|\mathcal{S}_x|} \sum_{j=1}^{|\mathcal{S}_x|} r_j$ 
18:   Downsample  $\mathcal{S}_x$  to a fixed update group  $\mathcal{B}_x$  of size  $n$ , trying to ensure  $\geq n/2$  size for each correct or incorrect subset when possible
19:   Compute the inclusion probability  $q_j$  for each retained sample  $(a_j, r_j) \in \mathcal{B}_x$  under the down-sampling scheme
20:   for each retained sample  $(a_j, r_j) \in \mathcal{B}_x$  do
21:     Compute IS-corrected advantage:  $A_j \leftarrow \frac{r_j - \hat{p}_x}{q_j}$ 
22:     Add  $(x, a_j, A_j)$  to the final batch:  $\mathcal{B} \leftarrow \mathcal{B} \cup \{(x, a_j, A_j)\}$ 
23:   end for
24: end for
25: Compute the policy gradient objective with batch  $\mathcal{B}$  and update the model.

```

Algorithm 2 REINFORCE-ADA-EST under the log objective (one training iteration)

```

1: Input: Current policy  $\pi_\theta$ , batch of prompts  $\mathcal{D}$ , effective update size  $n$ , sampling bounds  $[N_{\min}, N_{\max}]$ , total budget target  $N_{\text{total}}$ , and estimator  $\mathcal{E}$ 
2: Get current pass-rate estimates  $\hat{p}_x \leftarrow \mathcal{E}(x)$  for all  $x \in \mathcal{D}$ 
3: Allocate budgets  $N_x \propto 1/\sqrt{\hat{p}_x + \epsilon}$  subject to  $N_x \in [N_{\min}, N_{\max}]$  and  $\sum_x N_x \approx N_{\text{total}}$ 
4: for each prompt  $x \in \mathcal{D}$  do
5:   Sample  $N_x$  responses  $\{a_j, r_j\}_{j=1}^{N_x} \sim \pi_\theta(\cdot | x), r^*(\cdot, \cdot)$ 
6:   Compute the full-pool mean  $\bar{r}_x \leftarrow \frac{1}{N_x} \sum_{j=1}^{N_x} r_j$ 
7:   Apply the residual weight  $\alpha_x \leftarrow 1/\sqrt{\bar{r}_x + \epsilon}$ 
8:   Form weighted advantages  $A_j \leftarrow (r_j - \bar{r}_x)\alpha_x$ 
9: end for
10: Update the policy using the weighted batch and refresh  $\mathcal{E}$  with the new observations

```

C.2 Implementation Details and Variants of REINFORCE-ADA-EST

In practice, REINFORCE-ADA-EST follows an estimate-then-allocate pipeline. Given current pass-rate estimates $\hat{p}_x$, we allocate additional inference budget according to the target weighting, then construct a weighted update from the enlarged sample pool. Algorithm 2 shows the hybrid realization used in our main experiments under the log objective.

We consider two practical estimators for the pass rate. The first is a lightweight value model $V_\phi(x)$ that predicts prompt-level success probability. The second is ADA-EMA, a model-free Bayesian moving average inspired by Qu et al. (2025), which maintains exponentially decayed counts of total trials and positive outcomes for each prompt. In our main experiments, we use the hybrid log-objective strategy, allocating budget proportional to $1/\sqrt{p_i}$ and leaving the residual factor $1/\sqrt{p_i}$ in the gradient.

We also compare the effect of sampling and reweighting in REINFORCE-ADA-EST. The training reward averaged at the prompt level is similar for both the hybrid strategy ($w_{\text{sample}} = 1/\sqrt{p}, w_{\text{grad}} = 1/\sqrt{p}$) and the implicit weighting via sampling ($w_{\text{sample}} = 1/p, w_{\text{grad}} = 1$). However, the batch-level reward under implicit weighting eventually decreases. This indicates that the method is overly aggressive in sampling negative samples, leading to excessive oversampling of difficult prompts. When the task distribution contains many hard examples, this issue becomes more pronounced, since simpler samples receive too few draws.

In fact, the number of samples drawn per prompt N_x , the total sampling budget N_{total} , and the reward r_x satisfy

$$(r_{\text{batch}} - r_{\text{prompt}})N_{\text{total}} = \text{Cov}(N_x, r_x),$$

A large gap between prompt level reward and batch level reward implies a strong negative covariance between N_x and r_x , meaning low reward prompts dominate the sampling budget. Under a finite sampling constraint, it is therefore important to ensure broad prompt coverage and use a more moderate sampling strategy to avoid such imbalance and extreme behavior.

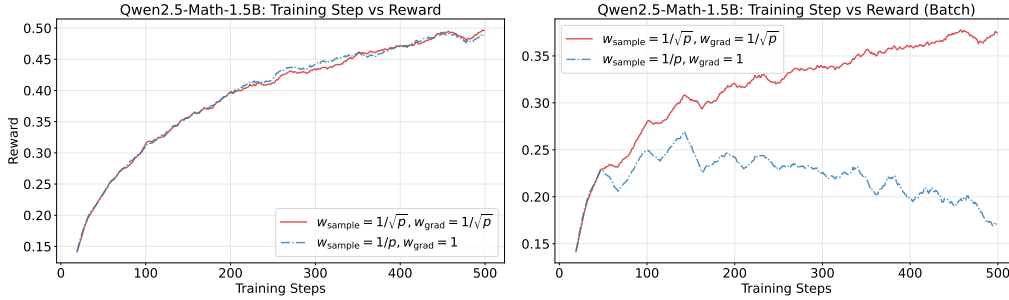


Figure 5: Comparison with different sampling/reweighting strategy.

D Variance-Reduction Gradient Estimator under the Non-linear RL Objective

In this section, we connect our general framework to the variance-reduction principle explored in Yao et al. (2025). We begin with the canonical log objective, generalize their findings from the specific case of Rejection Sampling Fine-Tuning (RFT) to the general Reinforce algorithm with a baseline, and then extend the same analysis to other non-linear objectives.

We recall that Yao et al. (2025) demonstrated that RFT can be viewed as maximizing the log-objective $J_{\log}(\theta) = \mathbb{E}[\log p_\theta(x)]$. Here, we analyze the gradient estimator for this objective under the general Reinforce algorithm with baseline:

$$g_{\log}(x) = \frac{1}{p(x)} \cdot \mathbb{E}_{a \sim \pi_\theta(\cdot|x)} \left[\nabla_\theta \log \pi_\theta(a|x) \cdot (r(x, a) - p(x)) \right]. \quad (8)$$

D.1 Optimal Budget Allocation

Our goal is to estimate the total gradient for a batch $g_{\text{batch}} = \sum_i g_{\log}(x_i)$ with minimum variance, given a total budget N . Suppose we have a total of N samples to distribute

across B prompts $x_1, \dots, x_B$. Using n_i samples for prompt x_i , the batch gradient estimator is

$$\hat{g}_{batch} = \frac{1}{B} \sum_{i=1}^B \frac{1}{n_i p_i} \sum_{j=1}^{n_i} \nabla_{\theta} \log \pi(a_{ij}|x_i) (r_{ij} - p_i), \quad (9)$$

where $p_i = p_{\theta}(x_i)$.

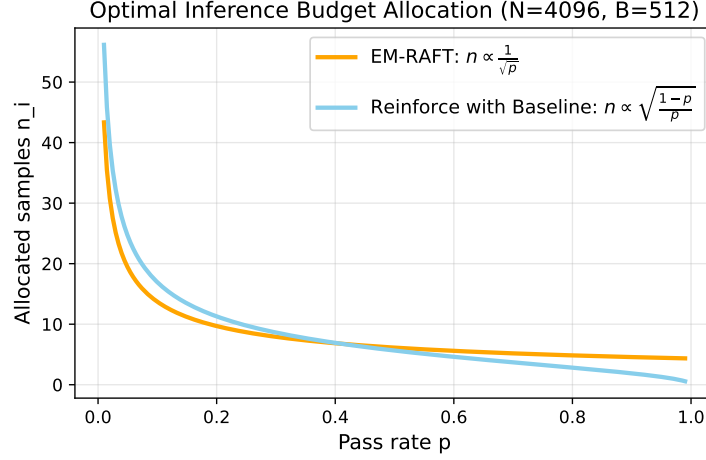


Figure 6: The theoretically optimal inference budget allocation of RAFT and Reinforce-with-baseline algorithms under the log objective.

Crucially, note the distinction from Section 2: In this estimator, the optimization of the log-objective is captured entirely by the explicit factor $1/p_i$. Since the inner sum is normalized by n_i , the allocation n_i does not affect the optimization objective (it is an unbiased estimator for any $n_i \geq 1$). Therefore, the role of n_i here is purely to minimize the variance of the estimator. We seek the allocation $(n_1, \dots, n_B)$ minimizing the total gradient variance:

$$\begin{aligned} \min_{n_1, \dots, n_B} \quad & V = \sum_{i=1}^B \frac{\sigma_g^2(x_i)}{n_i p_i^2}, \\ \text{subject to} \quad & \sum_{i=1}^B n_i = N \quad \text{and} \quad n_i \geq 1. \end{aligned} \quad (10)$$

Here $\sigma_g^2(x_i)$ is the variance of the baselined Reinforce gradient $\nabla_{\theta} \log \pi(a_{ij}|x_i)(r_{ij} - p_i)$. To make its dependence on $p(x)$ explicit, write $g(a) = \nabla_{\theta} \log \pi_{\theta}(a|x)$ and note that $r \in \{0, 1\}$ with $r \sim \text{Bernoulli}(p(x))$. For RFT without a baseline, the one-sample estimator is $g(a)r$, so $\mathbb{E}[\|g(a)r\|^2] = p(x)\mathbb{E}[\|g(a)\|^2 | r = 1]$. For Reinforce with a baseline, the one-sample estimator is $g(a)(r - p(x))$. If the positive and negative samples have the same conditional expected squared gradient norm,

$$\mathbb{E}[\|g(a)\|^2 | r = 1] \approx \mathbb{E}[\|g(a)\|^2 | r = 0] \approx C^2,$$

then

$$\begin{aligned} \mathbb{E}[\|g(a)(r - p)\|^2] &= p(1-p)^2 \mathbb{E}[\|g(a)\|^2 | r = 1] \\ &\quad + (1-p)p^2 \mathbb{E}[\|g(a)\|^2 | r = 0] \\ &\approx C^2 p(1-p). \end{aligned}$$

Therefore,

$$\sigma_g^2(x) \propto \begin{cases} p(x) & \text{(RFT without baseline),} \\ p(x) \cdot (1 - p(x)) & \text{(Reinforce with baseline).} \end{cases}$$

Solving Equation (10) via Lagrange multipliers gives the optimal allocation rule:

$$n_i^* \propto \frac{\sigma_g(x_i)}{p_i}, \quad (11)$$

which yields

$$\text{RFT: } n_i^* \propto \sqrt{\frac{1}{p_i}}, \quad \text{Reinforce: } n_i^* \propto \sqrt{\frac{1-p_i}{p_i}}.$$

This shows that incorporating a baseline reduces gradient variance and thus changes the optimal sampling strategy. Combining these findings, we can formulate a reasonable variance-reduction algorithm.

Reinforce with baseline. At each iteration t :

1. Allocate inference budgets as $n_i \propto N \cdot \sqrt{\frac{1-p_{\theta_t}(x_i)}{p_{\theta_t}(x_i)}}$.
2. For each prompt x_i , compute the advantage:

$$A^t(x_i, a_{ij}) = \frac{1}{n_i p_i} \cdot (r_{ij} - p_{\theta_t}(x_i)) \propto \frac{1}{\sqrt{p_{\theta_t}(x_i) \cdot (1 - p_{\theta_t}(x_i))}} \cdot (r_{ij} - p_{\theta_t}(x_i)).$$

3. Update the policy using the weighted gradient.

Remark 1 (Recovering the GRPO Advantage). *The derived advantage term $\frac{r-p}{\sqrt{p(1-p)}}$ is exactly equivalent to normalizing the reward by its standard deviation, since $\text{Std}[r] = \sqrt{p(1-p)}$ for binary rewards. This result provides an interesting explanation for the heuristic used in algorithms like GRPO (Shao et al., 2024), which normalize advantages by the group standard deviation.*

Remark 2 (The Necessity of the Log-Objective). *It is important to note that the log-objective factor $(1/p)$ in Equation (9) is essential for this result. If we were optimizing the standard objective $J = \mathbb{E}[p]$ (where the weight is 1), the variance term would be $\frac{\sigma_g^2(x_i)}{n_i}$ instead of $\frac{\sigma_g^2(x_i)}{n_i p_i^2}$. Minimizing that variance would lead to allocating more budget to prompts with maximum variance (i.e., $p \approx 0.5$) rather than the difficult prompts ($p \rightarrow 0$). Thus, the log-objective is a prerequisite for focusing the variance-reduction effort on difficult problems.*

Extension to power objectives. The same argument extends to the power objective $f(p) = p^c$ with weight $f'(p) = c p^{c-1}$. In that case,

$$V_c = \sum_{i=1}^B \frac{\sigma_g^2(x_i)}{n_i} c^2 p_i^{2c-2}, \quad n_i^* \propto c p_i^{c-1} \sigma_g(x_i).$$

Under the same conditional-gradient-norm assumption, this yields

$$\text{RFT: } n_i^* \propto c p_i^{c-\frac{1}{2}}, \quad \text{Reinforce: } n_i^* \propto c p_i^{c-\frac{1}{2}} \sqrt{1-p_i}.$$

This matches the discussion in Section 2: smaller c places more budget on low-pass-rate prompts, while larger c approaches more uniform allocation.

Extension to the logit objective. We also consider the two-sided objective implicitly optimized by REINFORCE-ADA-SEQ-BALANCE. As discussed in Section 2, the positive-only stopping rule corresponds to the log objective because sampling until the first positive sample yields an expected cost proportional to $1/p$. A natural symmetric extension is to keep sampling until both positive and negative evidence have been observed. In the simplest case, the expected stopping time then scales as

$$\frac{1}{p} + \frac{1}{1-p} - 1,$$

where the additive constant does not affect the induced weighting. This suggests the logit objective

$$f_{\text{bal}}(p) = \log \frac{p}{1-p}.$$

Its derivative is

$$f'_{\text{bal}}(p) = \frac{1}{p} + \frac{1}{1-p} = \frac{1}{p(1-p)}.$$

Substituting this into the general allocation rule $n_i^* \propto f'(p_i)\sigma_g(x_i)$ and using the same conditional-gradient-norm assumption, we obtain, for Reinforce with a baseline,

$$n_i^* \propto \frac{1}{p_i(1-p_i)} \sqrt{p_i(1-p_i)} = \frac{1}{\sqrt{p_i(1-p_i)}}.$$

This yields a symmetric U-shaped allocation rule: prompts receive more budget when either positive or negative evidence is rare. This matches the intuition behind REINFORCE-ADA-SEQ-BALANCE, which continues sampling until both sides of the signal have been observed.

E Simulation Studies for REINFORCE-ADA

To complement the full LLM experiments, we present a small set of simulation studies to better understand the behavior of REINFORCE-ADA-SEQ in isolation. The goal of this part is not to evaluate end-to-end model quality, but to build intuition for the sampling behavior induced by the sequential stopping rules themselves.

We begin with the sampling cost induced by these stopping rules, illustrated in Figure 7. We evaluate two variants of the method: REINFORCE-ADA-SEQ-POS, which continues sampling until K positive responses are observed, and REINFORCE-ADA-SEQ-BALANCE, which stops once both K positive and K negative responses have been collected.

For each strategy, we consider three maximum exploration budgets $N_{\text{max}} \in \{32, 64, 128\}$ and use a batch size of $k_{\text{batch}} = N_{\text{max}}/8$. We then vary the stopping requirement K and estimate the expected number of samples required as a function of the true success probability $p \in (0, 1)$. Because each response is a Bernoulli trial, the total number of required samples can be simulated accurately and efficiently.

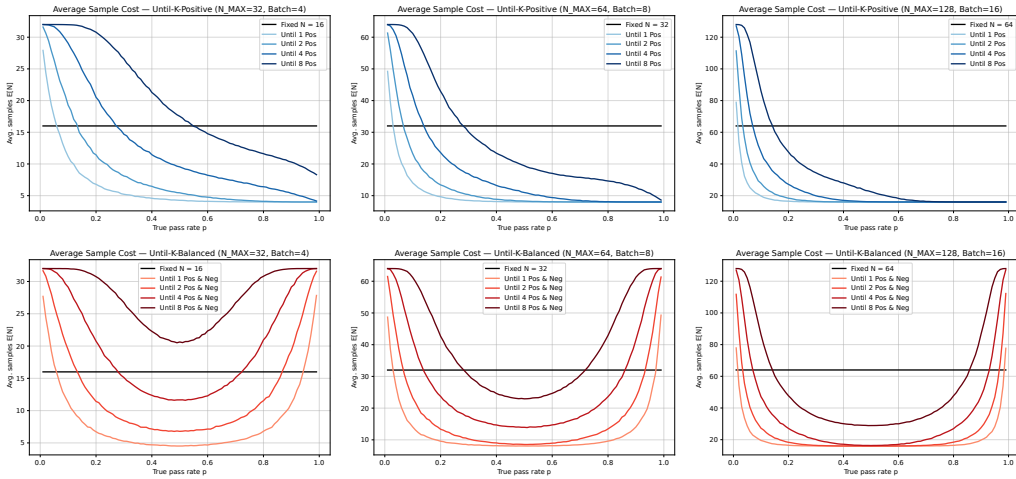


Figure 7: Estimated sampling cost under sequential stopping rules. The first row shows the expected number of responses required until observing K positive samples; the second row shows the cost until collecting K positive and K negative samples. Each column corresponds to a different maximum exploration budget $N_{\text{max}} \in \{32, 64, 128\}$.

Estimation Bias. The same simulation framework also lets us examine a second question: how much bias is introduced by the empirical pass-rate estimates used in adaptive sampling. To isolate this effect, we assume that the prompt set has difficulty uniformly distributed over $[0, 1]$, and evaluate the estimators on 100 evenly spaced true pass rates.

For REINFORCE-ADA-EST, we first use n samples to estimate the pass rate for each prompt. We then assume that the second stage has a total budget of $4 \cdot n \cdot 100$ samples, which is allocated proportionally to $\sqrt{(1-p)/p}$, where p is the first-stage estimated pass rate. To avoid extreme allocation, we additionally impose a clipping threshold of $15n$ samples per prompt.

For REINFORCE-ADA-SEQ, we sample sequentially in batches of size n , for at most 8 rounds. The REINFORCE-ADA-SEQ-POS variant stops once $n/2$ positive samples have been collected, while REINFORCE-ADA-SEQ-BALANCE stops once both $n/2$ positive and $n/2$ negative samples have been collected.

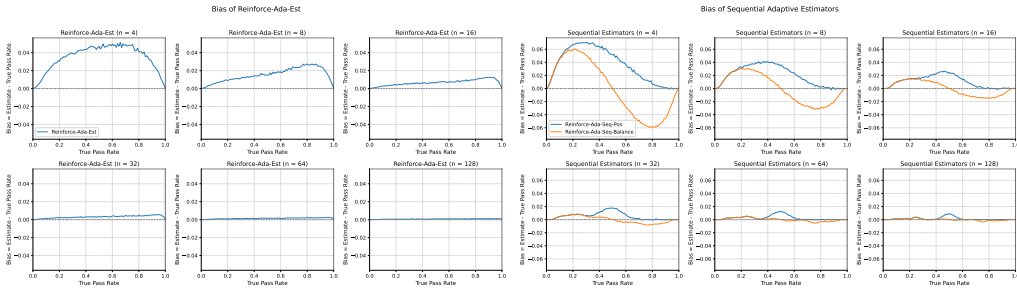


Figure 8: Bias of pass-rate estimation under adaptive sampling. Left: REINFORCE-ADA-EST, where a first-stage estimate based on n samples is followed by second-stage allocation proportional to $\sqrt{(1-p)/p}$ under a total budget of $4 \cdot n \cdot 100$ and a per-prompt cap of $15n$. Right: REINFORCE-ADA-SEQ, where sampling proceeds in batches of size n for up to 8 rounds, with POS stopping after collecting $n/2$ positive samples and BALANCE stopping after collecting both $n/2$ positive and $n/2$ negative samples. In both cases, the true prompt difficulty is assumed to be uniformly distributed over $[0, 1]$ and evaluated on 100 grid points.

The results in Figure 8 show that the induced bias is still substantial when the group size is small. For $n \in \{4, 8, 16\}$, the bias remains fairly large, and for $n = 4$ there is a considerable range of the true pass-rate spectrum where the estimation bias exceeds 0.04. In contrast, once we scale the group size to $n \geq 32$, the bias becomes much smaller, especially for REINFORCE-ADA-EST and REINFORCE-ADA-SEQ-BALANCE. This suggests that although our adaptive procedures do introduce estimation bias, that bias naturally shrinks as we scale up the group size.

F Additional Experiment Details

Models and benchmarks. We evaluate the generality of our framework across varying scales and architectures, using three foundation models: **Qwen2.5-Math** (1.5B and 7B) and **Llama-3.2-3B-Instruct**. For evaluation, we employ a comprehensive suite of mathematical reasoning benchmarks: **MATH500** (Hendrycks et al., 2021), **Minerva Math** (Lewkowycz et al., 2022), and **OlympiadBench** (He et al., 2024). Furthermore, to test robustness on competition-level problems, we compile an **AIME-like** test set consisting of 230 problems sourced from recent competitions: AIME24, AIME25, HMMT24, HMMT25, BRUMO25, AMC23, and CMIMC25 (Balunović et al., 2025). All evaluations report the **Pass@1** accuracy averaged over 32 samples (Ave@32), generated with a temperature of 1.0 and a max token limit of 4096. We also include a broader analysis of the reward-entropy trade-off and Pass@k behavior in Appendix G.

Training data and verifier. Training utilizes the standard subset of **OpenR1-Math-220k**³. We employ the **Math-Verify** tool (Kydlíček) for automatic solution correctness verification. For the main table, we use Qwen-Math verifier for Minerva math dataset for fair comparison, which has better extraction quality. To ensure a high-quality training signal, we implement a standard preprocessing pipeline: (1) Deduplication of prompts; (2) Exclusion of prompts where reference solutions fail verification; (3) **Difficulty-based Filtering**: Following Yang et al. (2024), we filter prompts based on empirical difficulty estimated by sampling 16 responses from the base model. We discard trivial prompts (average reward > 0.375) and effectively impossible prompts (0 correct responses). This helps us to focus the training distribution on problems of moderate difficulty rather than trivially easy or impossibly hard ones. This choice is motivated by the observation of Hugging Face (2025) that incorporating overly simple problems into the training set can degrade model performance.

RL training details. All experiments are conducted based on the ver1 framework (Sheng et al., 2024), where a Tinker-based⁴ implementation is also provided. Training is configured with a prompt batch size of 512, and each collected batch is used for four gradient-update steps. We use the AdamW optimizer with a fixed learning rate of 1×10^{-6} . To encourage exploration (with 10-step warm-up), an entropy regularization term with coefficient 1×10^{-4} is applied, while no KL penalty is introduced. Following Yu et al. (2025), we adopt the clip-higher trick by setting the clipping range to $[0.2, 0.28]$. For all variants, we keep these standard RL hyperparameters (learning rate, batch size, entropy coefficient, optimizer, etc.) fixed. No tuning was performed for REINFORCE-ADA. Each RL run is carried out for 600 training steps to obtain the final model.

Sampling configuration. The downsampling mechanism and random samples of REINFORCE-ADA-SEQ make the methods not directly comparable. We standardize the **update group size** (the number of samples used for the backward pass) to $n = 4$ for REINFORCE-ADA-SEQ and baseline GRPO, but we also include GRPO baselines with larger group size 8 and 16. We refer readers to Figure 11 for a sense of the sampling dynamics of different methods. Specifically, we consider the following competitors:

- **GRPO baselines:** We compare against GRPO with fixed group sizes $n \in \{4, 8, 16\}$.
- **REINFORCE-ADA-SEQ:** The algorithm adaptively samples up to $N_{\max} = 32$ responses per prompt but downsamples the collected pool to a fixed group size of $n = 4$ for the gradient update. This ensures that the training cost of the backward pass remains identical to GRPO- $n4$, with overhead limited only to inference.
- **REINFORCE-ADA-EST:** The total sampling budget is allocated proportionally to $1/\sqrt{\hat{p}_i}$. The hyperparameter value 8 is chosen to roughly match the sampling cost of REINFORCE-ADA-SEQ.

In general, REINFORCE-ADA-EST is most directly comparable to GRPO- $n8$, as they consume approximately the same number of forward and backward samples. In contrast, REINFORCE-ADA-SEQ involves inherent stochasticity in its adaptive sampling process, so only coarse-grained comparisons are possible; nevertheless, our experiments consistently show clear and robust trends across different models. Table 1 in the main text provides an illustration based on the Qwen2.5-Math-1.5B experiments.

G Additional Experiment Results and Ablation Studies

G.1 Adaptive Sampling Solves More Difficult Prompts

The impact of prompt set difficulty After ~ 200 steps (Fig. 11, right, second row), most prompts become easy—two rounds already satisfy the positive quota—so additional sampling brings diminishing returns. To stress-test our method, we construct a hard subset that

³<https://huggingface.co/datasets/open-r1/OpenR1-Math-220k>

⁴<https://github.com/thinking-machines-lab/tinker-cookbook>

Model	Algorithm	Math500	Minerva Math	Olympiad Bench	AIME-like	Weighted Average
<i>Qwen2.5-Math-1.5B</i>	GRPO-n4	74.2	34.4	38.4	16.2	45.3
	REINFORCE-ADA-SEQ-POS	75.8	35.7	38.6	16.5	46.1 (+0.8)
	REINFORCE-ADA-SEQ-BALANCE	77.4	36.5	40.5	17.5	47.6 (+2.3)
<i>Qwen2.5-Math-1.5B (hard)</i>	GRPO	71.0	31.8	34.3	13.8	41.9
	REINFORCE-ADA-SEQ-POS	73.9	33.1	36.4	16.4	44.6 (+2.7)
	REINFORCE-ADA-SEQ-BALANCE	74.7	33.7	38.7	17.6	45.5 (+3.6)
<i>Qwen2.5-Math-7B</i>	GRPO-n4	82.5	45.0	45.2	22.3	53.0
	REINFORCE-ADA-SEQ-POS	82.7	45.1	46.7	23.7	54.2
	REINFORCE-ADA-SEQ-BALANCE	84.0	45.2	47.1	23.7	54.6
<i>Qwen2.5-Math-7B (hard)</i>	GRPO	80.7	42.8	42.9	21.8	51.3
	REINFORCE-ADA-SEQ-POS	82.4	44.8	45.0	22.2	53.0
	REINFORCE-ADA-SEQ-BALANCE	83.2	45.0	46.9	25.2	54.3
	REINFORCE-ADA-EST	84.1	44.0	48.9	25.4	55.3

Table 3: Ablation study on the impact of prompt difficulty. The gains become more pronounced on the more challenging training set, where our methods allocate more inference budget to difficult prompts to recover a valid learning signal, while GRPO samples all prompts uniformly.

Method	Extremely Hard	Hard	Medium	Easy	Overall
Base Model	0.0000	0.0889	0.2950	0.6151	0.1139
GRPO	0.3414 (+34.14)	0.4639 (+37.51)	0.6497 (+35.46)	0.6865 (+7.14)	0.4749 (+36.11)
DAPO	0.3487 (+34.87)	0.4648 (+37.60)	0.6521 (+35.71)	0.5580 (-5.70)	0.4757 (+36.19)
Reinforce-Ada	0.3674 (+36.74)	0.4825 (+39.37)	0.6579 (+36.29)	0.7153 (+10.02)	0.4933 (+37.94)

Table 4: Accuracy and absolute improvements over the base model across difficulty levels. SEQ-BALANCE achieves consistent improvements across all difficulty categories.

keeps only prompts with 1-2 correct out of 16 initial samples. As shown in Fig. 9, adaptive sampling yields a much larger margin over GRPO on this challenging set, and the gap widens late in training. This is also reflected on the final model performance reported in Table 3.

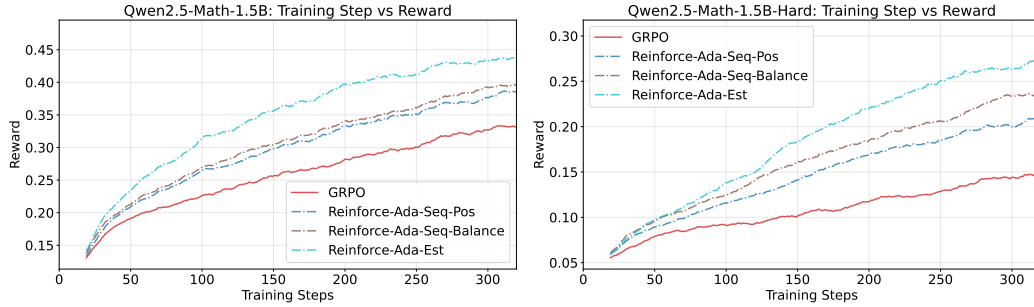


Figure 9: Ablation study on the prompt set difficulty. Left: prompt set with moderate difficulty. Right: challenging prompt set. The benefit of adaptive sampling is more obvious with challenging prompt set.

Intuitively, harder prompts reduce all-correct saturation and keep uncertainty high; REINFORCE-ADA continues to mine informative negatives and avoids premature deactivation, sustaining exploration and gradient signal.

To better understand how adaptive allocation reshapes the learning dynamics, we randomly sample a subset of 5000 prompts from the hard training set. We partition the 5000-prompt subset into four difficulty levels using the base model’s pass@16: (i) Extremely Hard (0), (ii) Hard (0–0.2], (iii) Medium (0.2–0.5], and (iv) Easy (0.5–1.0]. The dataset is dominated by difficult items where 984 (19.7%) are extremely hard and 3,082 (61.7%) are hard. Meanwhile, medium and easy prompts account for 870 (17.4%) and 63 (1.3%), respectively. As expected, the base model’s accuracy rises monotonically as prompts become easier, achieving 0.00%, 8.89%, 29.50%, and 61.51% on the four groups.

Table 4 summarizes different models’ accuracy across four difficulty levels. All three RL methods substantially improve over the base model, but REINFORCE-ADA-SEQ-BALANCE achieves the highest accuracy in every category. The gains are especially pronounced on the Extremely Hard and Hard prompts—where the base model succeeds rarely or not at all—improving accuracy by +36.74 and +39.37 points, respectively. REINFORCE-ADA-SEQ-BALANCE also attains the best performance on Easy prompts (71.53%), whereas DAPO even hurts the performance in this regime. A likely reason is that DAPO tends to discard high-pass-rate prompts during training, while our balanced adaptive sampling variant allocates sufficient inference budget to preserve learning signals even for easy items.

G.2 Training Reward Curves Across Backbones

Please refer to Figure 10.

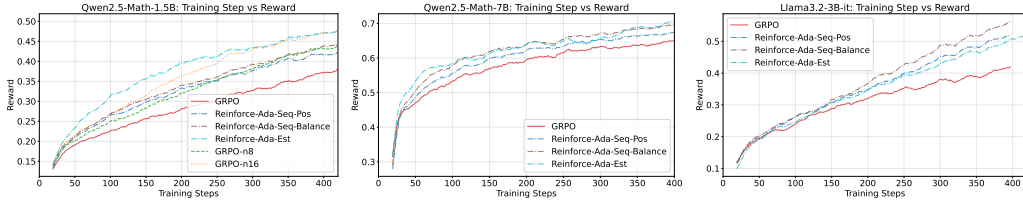


Figure 10: Training reward vs. steps for GRPO and REINFORCE-ADA across backbones: Qwen2.5-Math-1.5B, Qwen2.5-Math-7B, and Llama-3.2-3B-it. Curves are smoothed with a 20-step moving average. In these experiments, REINFORCE-ADA generally learns faster and reaches a higher reward than GRPO.

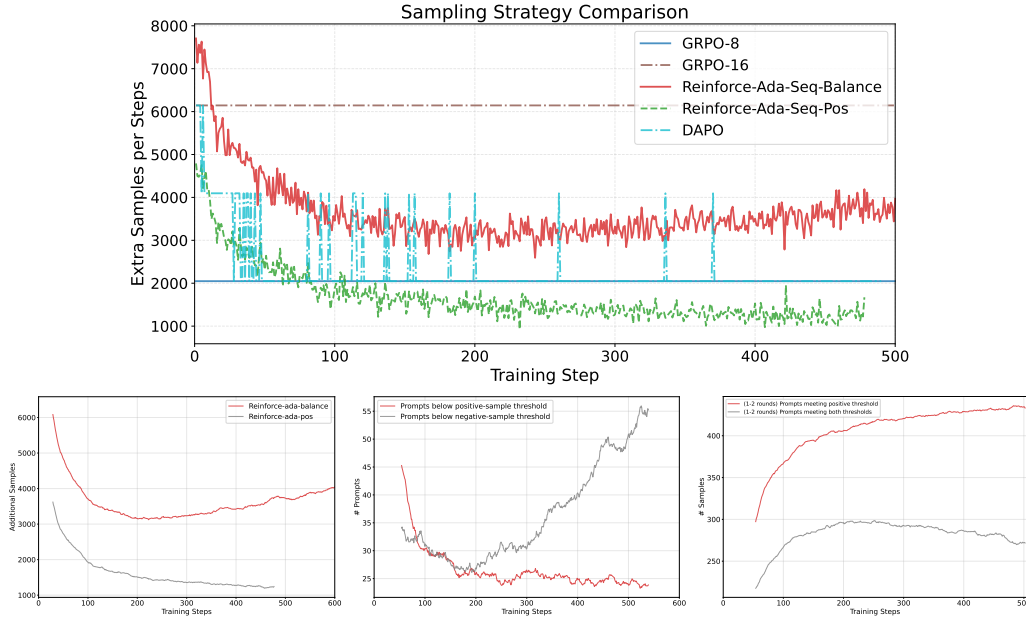


Figure 11: First row: Sampling dynamics of different training strategies using the Qwen2.5-Math-1.5B model. We omit REINFORCE-ADA-EST since its sampling cost matches that of GRPO-8. Second row: Sampling dynamics with the Qwen2.5-Math-1.5B model. Left: additional samples generated in later rounds compared to standard GRPO. Middle: number of prompts that remain active after multi-round adaptive sampling with the REINFORCE-ADA-SEQ-BALANCE variant. Right: number of prompts that satisfy the stopping criteria within the first two rounds with the REINFORCE-ADA-SEQ-BALANCE variant. All curves are smoothed using a moving average with a window size of 20.

G.3 Sampling Dynamics and Inference Cost

Inference Cost Profile. The first row of Figure 11 summarizes the *inference cost*, measured as the average number of generated samples per iteration.⁵ This view is intended to characterize the sampling behavior of different methods rather than compare their final performance. The main pattern is that GRPO-n4 has the lowest inference cost, SEQ-POS adds moderate overhead, SEQ-BALANCE uses more inference because it continues sampling to recover both positive and negative evidence, and GRPO-n16 remains the most expensive because it pays for a uniformly large group on every prompt. This illustrates the basic allocation trade-off behind adaptive sampling: it spends additional inference only where extra evidence is needed, instead of paying the full cost of large-group sampling everywhere.

The Sampling Dynamics of ADA-SEQ. Figure 11 provides a deep dive into how ADA-SEQ allocates its budget over time in the Qwen2.5-Math-1.5B experiments:

- **Early Training (Exploration Phase):** The cost is high as the model struggles with difficult prompts. The algorithm automatically allocates more samples to find the sparse positive solutions needed to initiate learning.
- **Mid Training (Consolidation Phase):** As the model improves, positive samples become easier to find. The sampling cost for SEQ-POS drops monotonically.
- **Late Training (Refinement Phase):** Crucially, SEQ-BALANCE exhibits a distinct “U-shaped” cost curve. After ~ 300 steps, the cost rises again. This phenomenon occurs because the model becomes highly proficient, making *negative* samples (failures) rare. To maintain a valid gradient signal (variance > 0), the algorithm actively hunts for these hard negatives. This adaptivity prevents the “all-correct” signal collapse that plagues standard GRPO in later stages.

G.4 Ablating Standard-Deviation Normalization in Scaled ADA-SEQ-BALANCE

We next study whether the scaled ADA-SEQ-BALANCE configuration should also include the GRPO-style standard-deviation normalization in the final update. The experiment is conducted on Qwen2.5-Math-1.5B using the smaller scaling configuration only: maximum inference size 32, sub-sampled training group size 8, and stopping thresholds $K_{\text{pos}} = K_{\text{neg}} = 4$. We report this setting separately because it is part of the scaling study and is intended to trace a compute frontier rather than match the default fixed-budget benchmark.

This ablation asks whether, beyond using the utility view to drive adaptive allocation, we should also explicitly normalize the update by $1/\text{std}$, which moves the training signal toward the GRPO-style non-linear objective associated with $2 \arcsin(\sqrt{p})$. Figure 12 shows that this extra normalization is consistently harmful in this setting. The variant with standard-deviation normalization performs worse than the unnormalized ADA-SEQ-BALANCE from pass@1 to pass@16, with the plotted pass@1, pass@4, and pass@8 curves all clearly below the corresponding baseline. This result suggests that directly coupling adaptive allocation with the non-linear objective can be detrimental here, which is consistent with our discussion in the main text about the risk of tightly coupling allocation rules to a specific objective when the primary quantity of interest is pass rate itself.

G.5 Entropy Dynamics and Pass@k

Beyond simple accuracy, we evaluate the quality of the learned policy through the lens of the reward–entropy trade-off. A known challenge in evaluating the reasoning ability of LLMs is the trade-off between reward (accuracy) and entropy (generation diversity). Previous work has argued that post-training can trade this uncertainty for higher rewards in a predictable manner (Cui et al., 2025). Moreover, prior work shows that standard GRPO

⁵We remark that GRPO-n16 uses more samples per iteration, resulting in a higher backward-pass cost.

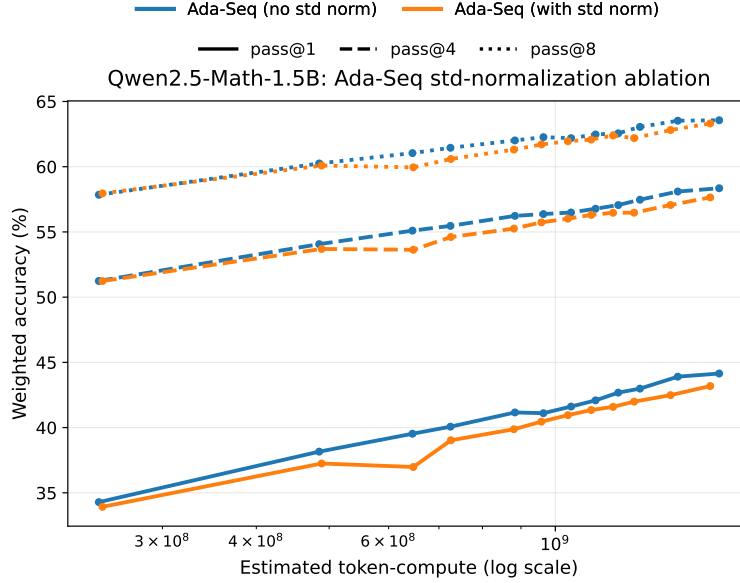


Figure 12: Ablation of GRPO-style standard-deviation normalization in the scaled ADA-SEQ-BALANCE setting on Qwen2.5-Math-1.5B. The compared configuration uses maximum inference size 32, sub-sampled training group size 8, and stopping thresholds $K_{\text{pos}} = K_{\text{neg}} = 4$. Adding the extra $1/\text{std}$ normalization degrades performance across the reported pass@1, pass@4, and pass@8 curves, and the same trend also holds through pass@16.

often collapses to a low-entropy policy early in training, reducing reasoning diversity and thus worsening pass@k for larger k (Shao et al., 2024; Yue et al., 2025).

An ideal algorithm should improve rewards without causing the policy to become overly deterministic (entropy collapse). To provide a more comprehensive evaluation, we analyze this trade-off here and present the results in Figure 13.

- (i) **Reward-entropy frontier.** While the precise entropy dynamics can vary depending on the foundation model, REINFORCE-ADA consistently achieves a comparable or superior reward-entropy curve. Specifically, from panel (2-1) of Figure 13, we observe that increasing the group size of GRPO does not improve the reward-entropy trade-off. On Qwen2.5-Math-1.5B (1-1), GRPO concentrates mass early (low entropy, narrow cloud) and achieves lower reward for a given entropy. All REINFORCE-ADA variants shift the frontier outward, and REINFORCE-ADA-SEQ-BALANCE lies furthest outward: at equal reward, it sustains higher entropy, and at equal entropy, it achieves higher reward. On Llama-3.2-3B-it (1-2), the base policy starts with a higher entropy floor, so the separation among different methods is smaller. Even so, the ADA-SEQ variants achieve a more favorable frontier than GRPO, and ADA-EST performs even better. We hypothesize that adaptive sampling encourages the model to explore prompts about which it is uncertain, thereby preserving entropy. We also observe that $\text{REINFORCE-ADA-SEQ-BALANCE} > \text{REINFORCE-ADA-SEQ-POS}$. We attribute this to the fact that exposure to negative signals discourages the model from becoming over-confident in a single solution path, thus preserving valuable policy diversity.
- (ii) **Pass@k behavior.** A related observation is that moderate policy entropy typically converts into improved pass@k relative to the base model over a wide range of k . Specifically, panel (2-2) shows that all RL methods dominate the base across k , but the largest, most practical gains appear at small budgets ($k \leq 8$). In this regime, REINFORCE-ADA yields the highest Pass@k; the advantage narrows as k grows (diminishing-returns regime), where REINFORCE-ADA-SEQ-BALANCE typically remains marginally best. The pattern implies two complementary effects: (a)

improved top-1 quality (higher reward at a given entropy), and (b) retained diversity among high-scoring modes (shallower saturation with k). Together, adaptive sampling moves the reward–entropy curve outward and converts that shift into higher Pass@ k at realistic attempt budgets, with REINFORCE-ADA-SEQ-BALANCE offering the most stable trade-off.

We also present the corresponding reward–entropy comparisons for the scaled configurations used in the compute-frontier study. Figure 14 shows that the same qualitative conclusion continues to hold after scaling up: ADA-SEQ-BALANCE improves the pass-rate–entropy frontier relative to the matched GRPO baselines. In other words, the gain from adaptive allocation is not limited to the default setting; even in the higher-compute regime, it attains higher pass rate at comparable entropy and maintains higher entropy at comparable pass rate.

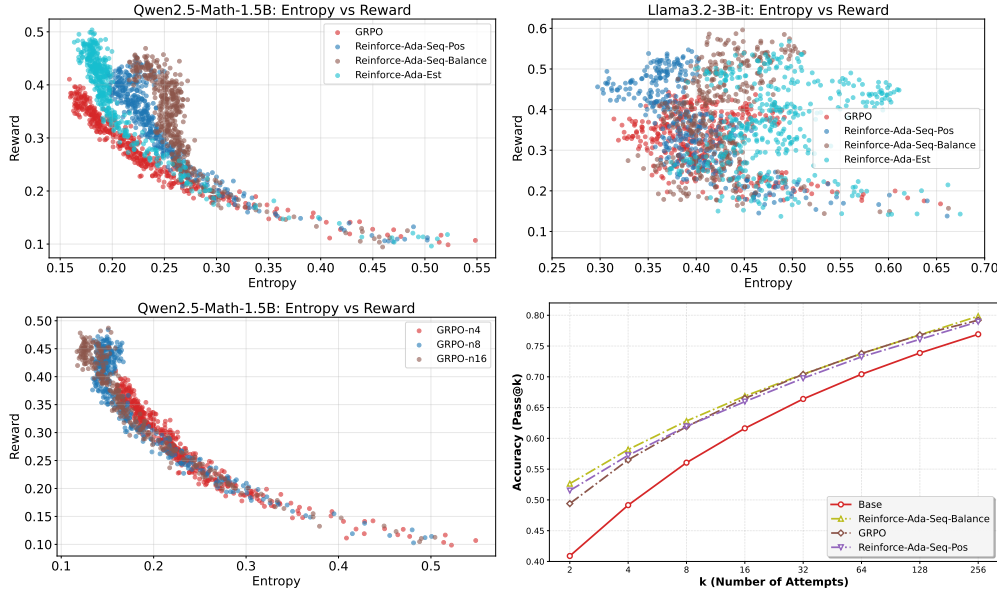


Figure 13: Reward–entropy trade-off in the first three panels and Pass@ k in the bottom-right panel on the test benchmarks. The three reward–entropy panels correspond to Qwen2.5-Math-1.5B, Llama-3.2-3B-it, and an additional Qwen2.5-Math-1.5B comparison under the difficulty ablation setting. Across these views, REINFORCE-ADA shifts the frontier outward, achieving higher reward at fixed entropy and higher entropy at fixed reward, and converts this into stronger Pass@ k at small, practical budgets ($k \leq 8$), with REINFORCE-ADA-SEQ-BALANCE typically best.

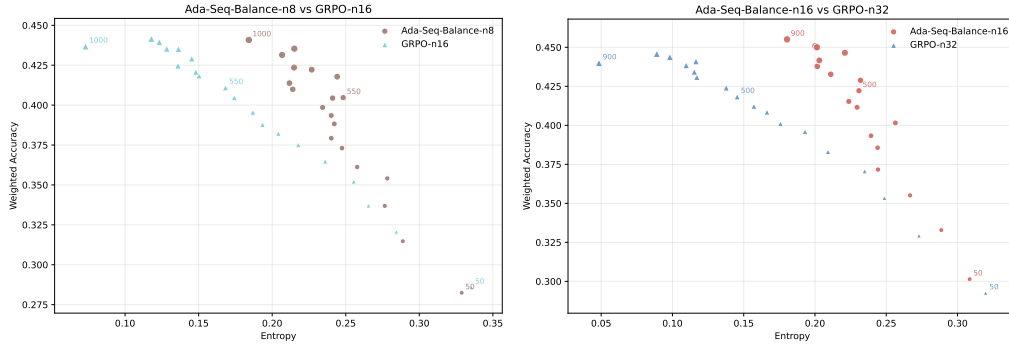


Figure 14: Reward–entropy comparisons for the scaled configurations on Qwen2.5-Math-1.5B. Left: ADA-SEQ-BALANCE with keep size 8 and maximum inference size 32 versus GRPO-n16. Right: ADA-SEQ-BALANCE with keep size 16 and maximum inference size 64 versus GRPO-n32. In both regimes, adaptive sampling improves the pass-rate–entropy frontier relative to the matched GRPO baseline.