Reward Budgeting Reduces Premature Convergence in Reinforcement Learning for LLM Reasoning

Mengni Jia[2] **Mengyu Zhou**[1,†] **Xiaoxi Jiang**[1] **Guanjun Jiang**[1]
[1]Qwen Business Unit, Alibaba [2]University of Cambridge
[†]Corresponding author: zhoumengyu.zmy@alibaba-inc.com

Abstract

RL for LLM reasoning often plateaus early as its policy sharpens. Policy entropy is studied as both a diagnostic and a control signal, but increasing it alone does not reliably change the eventual plateau in our experiments. We analyze RL's implicit target policy using a sharpness factor κ and effective counterpart κ_{eff} that describe how reward or advantage gaps translate into probability-ratio gaps. We introduce *reward budgeting*, a plug-in advantage construction assigning a fixed total reward budget to each prompt. For binary rewards and independent on-policy rollouts, its expected unclipped score-function update is the exact gradient of a finite-G concave objective with diminishing returns; its advantage gap decreases as more rollouts become correct. For general rewards, a shifted log-mean interpretation holds asymptotically under the stated regularity conditions. On Qwen3-8B and Llama3.2-3B-Instruct, reward budgeting consistently improves RL baselines, with gains up to +12.50 on AMC23 and +10.00 on AIME24 and AIME25. These results support an update-dynamics view of premature convergence and substantial gains without explicitly preserving entropy. Code is available at `https://github.com/Qwen-Applications/RewardBudget`.

1 Introduction

Policies trained with RL for LLM reasoning often improve rapidly at first and then enter a plateau where further training brings little gain. Such a plateau need not reflect an intrinsic limit of the model or data: optimization choices can change the performance reached under otherwise similar training conditions. For example, [12] report that, in their scaling study, GSPO [35] and CISPO [18] reach higher fitted asymptotic performance than DAPO [31]. This motivates us to distinguish *natural convergence* from **algorithm-induced premature convergence**: training may stop making useful progress, even when a different algorithm can reach a better solution.

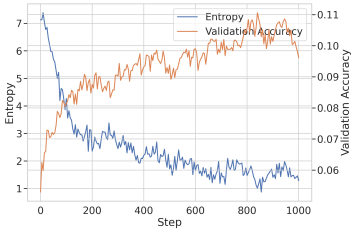


Figure 1: Average training entropy and validation accuracy across 12 Llama3.2-3B-Instruct runs.

This phenomenon has been commonly referred to as "entropy collapse". This name arises from the **observation** that the policy seems to stop learning and converge prematurely once its entropy drops to a relatively low level [3]; see Figure 1. Consequently, many proposed remedies focus on *explicit* entropy control, typically by encouraging higher entropy or keeping it within a prescribed range [3, 26, 2, 24, 10, 4, 7, 32, 25, 19, 28, 20, 5, 13, 33, 31, 22, 30, 32, 15, 11]; see §2 for a discussion. These studies provide substantial evidence that entropy is related to RL training dynamics. Here, we ask a narrower question: *is changing entropy itself sufficient to change the eventual performance plateau?*

Our results suggest caution. First, the familiar entropy–accuracy pattern is often observed after averaging across runs, while individual methods have different entropy trajectories (§3.3). Second, in

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

the intervention study of §3.1, two different entropy-promoting interventions substantially change entropy during early training, yet the trajectories return toward the same baseline behavior after the interventions are removed and do not reach a better final plateau. This does not imply that entropy-based methods cannot be effective: such methods may alter optimization in useful ways beyond simply raising entropy. It does suggest that, in our setting, entropy promotion alone is not a sufficient intervention for premature convergence.

Conceptually, entropy measures how dispersed the policy distribution is, but it does not indicate whether the resulting responses are correct or whether their diversity is useful. A high-entropy policy can produce diverse but unproductive outputs, while a low-entropy policy can be confidently correct. We therefore view entropy as an informative diagnostic of policy concentration rather than, by itself, an explanation of why training plateaus. For this reason, we use the term *premature convergence* rather than *entropy collapse* throughout the paper.

Once we question whether entropy is the primary cause behind the plateau, a natural question follows: if we do not control entropy directly, how can we sustain meaningful learning signals so that training can continue to realize incremental gains? In fact, reduced entropy (and the resulting sharper policy distribution) is expected, since it indicates that the policy is learning and becoming more confident on the training data. We just wish to avoid improper sharpening that locks the policy into cheap pseudo-modes and leaves little room for further gains. In generative modeling, mode-seeking behavior can depend on the divergence being optimized [14]. This motivates examining objective-dependent sharpening in our RL setting. However, simply delaying convergence does *not* guarantee stronger performance: gentler dynamics can slow training and require more data for sufficient learning. What we need is a mechanism that maintains gentler training dynamics to prevent premature sharpening, while adaptively relaxing control when genuine learning signals appear.

To describe the training dynamics more directly, we analyze RL through its implicit target policy in §3. For KL-regularized RL, the optimal target takes the reward-weighted form $\pi^*(\gamma) \propto \pi_{\mathrm{ref}}(\gamma) \exp(R(\gamma)/\beta)$. This view isolates a *sharpness factor* κ, which determines how strongly reward gaps are amplified into probability-ratio gaps between higher- and lower-reward responses. We further define an effective version, κ_{eff}, for practical algorithms with normalized advantages, clipping, and other implicit constraints. In this sense, κ and κ_{eff} describe the sharpening pressure induced by the update itself, rather than the resulting uncertainty of the policy.

Motivated by this, we introduce **reward budgeting** (code repository). For each prompt, we allocate a fixed total reward budget across sampled rollouts according to their positive reward gaps. In the binary-reward setting, if k out of G rollouts are correct, each correct rollout receives a reward share proportional to $1/k$. As a result, the advantage gap decreases as a prompt becomes easier, producing stronger sharpening when successes are rare and gentler sharpening when successes are already common. Importantly, this behavior emerges without explicitly constraining policy entropy. We also discuss reference-free optimization in §4.2: removing a fixed KL reference avoids permanently anchoring the target policy to a static prior, while importance-ratio clipping can still provide local regularization.

Theoretically, reward budgeting admits a principled interpretation rather than a heuristic tweak. First, it regulates the κ trajectory, distinguishing it from most existing methods. As more rollouts become correct, each correct rollout receives a smaller share of the reward, which implicitly reduces the sharpness factor κ and automatically relaxes the sharpening pressure. Second, under binary rewards, which match our main RLVR experiments, the expected update of the proposed estimator is exactly the gradient of a finite-G concave objective with diminishing returns (§A.3). Thus, repeatedly solving an already-easy prompt contributes progressively less marginal benefit than under a linear expected-reward objective. Under general rewards, the gap-normalized estimator admits a shifted log-mean interpretation in the large-G limit under suitable support assumptions (§A.4).

Empirically, our method substantially improves on Qwen3-8B and Llama3.2-3B-Instruct across various RL baselines (up to +12.50 on AMC 23 and +10.00 on AIME24 & AIME25). The gains do not require higher entropy; after applying reward budgeting, entropy often decreases while final performance improves. Collectively, these results suggest that regulating algorithm-induced sharpening is a practical way to mitigate premature convergence, while entropy remains a useful diagnostic rather than a necessary quantity to preserve.

2 Related work

Premature convergence in RL for LLM reasoning is often accompanied by a rapid decline in policy entropy. Recent studies have examined this behavior from different perspectives, including entropy dynamics, token selection, reward shaping, and regularization. While these approaches share an interest in improving exploration, they differ in how entropy enters the policy update.

Entropy dynamics and token-level interventions. [3] establish an empirical relationship between policy entropy and validation performance, and analyze entropy changes through the covariance between action log-probabilities and policy updates. Their analysis motivates Clip-Cov and KL-Cov, which selectively constrain high-covariance token updates to slow entropy reduction. From another perspective, [26] find that a small fraction of high-entropy tokens act as important forks in reasoning. They show that restricting policy-gradient updates to these tokens can match or outperform full-token training. Both works highlight the importance of where and how policy updates are applied. However, they focus on different signals: covariance characterizes contributions to entropy change, while token entropy identifies uncertain decision points. Our analysis instead examines how the RL objective and advantage construction determine the sharpening pressure of policy updates.

Entropy-guided credit assignment. Other studies use entropy to improve the learning signal rather than directly regularizing the policy distribution. [2] associate high-entropy regions with exploratory reasoning behaviors, such as reflection and correction. They add a clipped, gradient-detached entropy term to token advantages, encouraging such behaviors without directly optimizing an entropy bonus. Similarly, [24] propose GTPO and GRPO-S to address coarse-grained credit assignment in GRPO. Their methods use entropy to assign greater rewards to uncertain decisions in successful responses and stronger penalties to confident errors, at the token and sequence levels, respectively. These approaches show that entropy can help identify where learning signals may be more useful. Our method also modifies reward allocation, but uses the relative rewards of rollouts within each prompt rather than their entropy. The goal is to adjust how strongly each prompt contributes to learning as its sampled success rate changes.

Selective entropy regularization. Direct entropy regularization provides another way to encourage exploration, but increasing entropy indiscriminately can also destabilize training. [10] study this issue and propose SIREN, which combines top-p action masking, high-entropy token selection, and a self-anchored regularizer. Rather than simply maximizing entropy, SIREN maintains a more controlled level of uncertainty over selected actions and positions. This distinction is important: the effectiveness of entropy-based methods can depend on which parts of the policy are regularized and how the updates are constrained, not only on the resulting entropy level.

Group-relative optimization and reward allocation. Our work is also related to the design of group-based RL objectives. GRPO [21] estimates advantages from relative rewards within each prompt group. Dr. GRPO [16] shows that group standard-deviation normalization can introduce difficulty-dependent weighting, and removes this normalization along with response-length normalization. DAPO [31] improves training through several changes, including dynamic sampling that excludes groups with uniformly correct or incorrect rollouts. These methods improve advantage estimation or group selection. In contrast, reward budgeting changes how rewards are allocated within each group. Instead of assigning each successful rollout an independent unit reward, it distributes a fixed budget across the group according to reward gaps. Under binary rewards, this makes the advantage gap decrease as the number of correct rollouts increases.

Relation to our work. The above studies show that entropy can be useful for understanding and improving RL, but its role depends on how it is used. We examine a related question: whether increasing entropy itself is sufficient to change the final performance plateau. Our intervention experiments suggest that this is not reliable in the settings we study. We therefore focus on the sharpening pressure induced by policy updates, characterized by κ and its effective counterpart κ_{eff}. Reward budgeting provides a simple way to adjust this pressure through advantage construction, without explicitly controlling entropy. Moreover, under binary rewards, its expected policy-gradient update corresponds to a finite-G concave objective with diminishing returns. Our results suggest that prompt-level reward allocation is another useful way to address premature convergence, alongside existing entropy-guided approaches.

3 Sharpness analysis

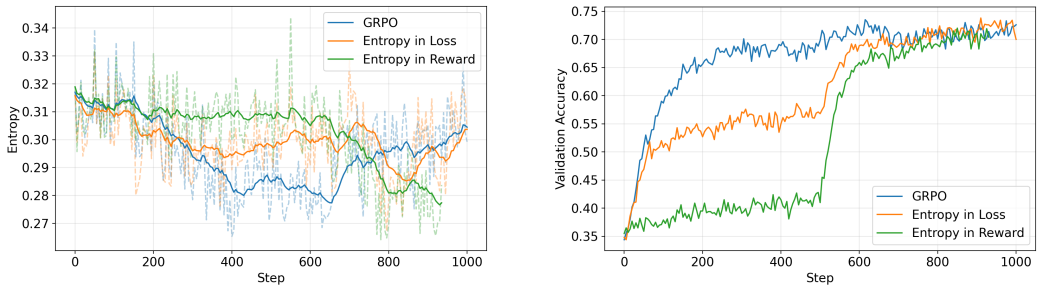
Premature convergence is common in RL training, but its underlying dynamics remain unclear. In this section, we examine policy sharpening as a more direct description of RL updates than entropy alone. In §3.1, we show that increasing entropy alone does not reliably improve the final performance plateau in our settings. In §3.2, we derive a sharpness factor κ from the RL objective, which characterizes how reward gaps are exponentially amplified in the implicit target policy. Finally, controlled experiments in §3.2.1 and individual-run analyses in §3.3 show how sharpening strength relates to convergence dynamics, while averaging can obscure inconsistencies in the entropy–accuracy relationship.

3.1 Entropy

Due to the observation that policies often plateau as their entropy drops, many remedies propose to control entropy. However, this correlation does not establish that entropy reduction causes the plateau. Before taking any action, we must first ask: is entropy truly the cause, or more like a correlate? This question is highly important, because treating the correlational signal as causal can lead to *ineffective* interventions. Entropy may instead serve as a useful diagnostic of convergence dynamics, without being a sufficient target for intervention. We test this hypothesis as follows.

Let $H(\pi_\theta)$ denote entropy on training data. We train Qwen3-8B in three runs that differ only by an *early entropy-based intervention*. For the first 500 steps, we use (1) **Baseline**: GRPO [21]. (2) **Add Entropy to Loss**: $J_{\mathrm{RL}}(\theta) + 0.01\, H(\pi_\theta)$, and (3) **Add Entropy to Reward**: $R(o) + 0.01 H(\pi_\theta(o))$. After 500 steps, *all* runs revert to the baseline and train for another 500 steps.

If early entropy promotion is sufficient to overcome premature convergence, the interventions should lead to higher post-intervention plateaus. Otherwise, the runs may return to similar performance once the interventions are removed.



(a) Entropy over training steps.

(b) Validation accuracy over training steps.

Figure 2: Entropy-intervention test: each method uses its own setting for the first 500 steps and then switches back to the same baseline for steps 500–1000.

From Figure 2a, both interventions increase entropy, yet neither improves final performance. Figure 2b shows no post-intervention after-effect: the trajectories return toward the baseline once the interventions are removed, with early intervention-runs even lagging behind initially. These interventions act through different mechanisms and increase entropy by different amounts, yet neither produces a lasting performance gain. This suggests that entropy is a useful diagnostic correlate of convergence dynamics, but increasing it alone is not sufficient to improve the final plateau in our settings. This does not rule out a causal role for entropy in other interventions.

Entropy View	Supported?	What it suggests
A Sufficient Causal Lever	✗	The tested early entropy interventions did not improve the final observed plateau.
A Diagnostic Correlate	✓	Entropy can indicate over-converging dynamics.

3.2 Sharpness factor

Consider the RL objective

$$J_{\mathrm{RL}}(\theta) = \mathbb{E}\big[R(\gamma)\big] - \beta\, D_{\mathrm{KL}}\big(\pi_\theta \,\|\, \pi_{\mathrm{ref}}\big),$$

where π_{ref} is the reference policy, $R(\gamma)$ is the reward of a trajectory γ, and $\beta > 0$ penalizes deviation from π_{ref}. Prior works often explain premature convergence by analyzing its local gradients. Here, we take a global view: this objective is equivalent to minimizing the *reverse KL* between π_θ and the **target policy** π^* (§A.6):

$$\pi^*(\gamma) \propto \pi_{\mathrm{ref}}(\gamma)\, \exp(R(\gamma)/\beta)\,, \tag{1}$$

This view makes the effect of RL transparent: it pushes π_θ toward a *reward-weighted* version of π_{ref}. On a finite response space, as $\beta \to 0^+$, this target concentrates on reward-maximizing responses within the support of π_{ref}. The limit is deterministic only when the supported maximizer is unique; with ties, it retains the reference policy's relative weights among the maximizers.

Consider two responses o^+ and o^- to a prompt p that receive a higher reward r^+ and a lower one r^-, respectively. From Eq 1, the relative probability that π^* assigns to o^+ over o^- is exponentially controlled by the **(principled) sharpness factor** κ:

$$\frac{\pi^*(o^+ \mid p)}{\pi^*(o^- \mid p)} = \frac{\pi_{\mathrm{ref}}(o^+ \mid p)}{\pi_{\mathrm{ref}}(o^- \mid p)} \exp\!\left(\frac{r^+ - r^-}{\beta}\right), \qquad \boxed{\kappa := \frac{r^+ - r^-}{\beta},} \tag{2}$$

For $\beta > 0$, κ controls the *multiplicative amplification* of higher- versus lower-reward response odds relative to the reference policy. For example, $\beta = 0.001$, $r^+ = 1$, and $r^- = 0$ give $\kappa = 1000$, so the target odds are the reference odds multiplied by $e^{1000} \approx 1.97 \times 10^{434}$. This characterizes the idealized target, rather than the speed with which a parameterized optimizer reaches it. Strong concentration can reduce probability assigned to lower-reward responses, some of which may contain useful reasoning segments.

The target-policy calculation above describes a KL-regularized objective. Practical algorithms also use normalized advantages and importance-ratio clipping, so their realized updates need not equal that idealized target. For a chosen higher- and lower-advantage response pair, we define an operational **effective sharpness factor** as the change in log odds:

$$\boxed{\kappa_{\mathrm{eff}}(s; a^+, a^-) := \log \frac{\pi_{\mathrm{new}}(a^+ \mid s)}{\pi_{\mathrm{new}}(a^- \mid s)} - \log \frac{\rho(a^+ \mid s)}{\rho(a^- \mid s)}.} \tag{3}$$

Here ρ is the policy before the update, and all four probabilities must be positive. In the reported measurements, each response is a full sequence: after updating a batch, we recompute log-probabilities for the same sampled responses and sum token log-probability changes relative to their stored rollout policy. For each prompt, we select one maximum-advantage and one minimum-advantage response and subtract their sequence log-probability changes. We skip groups with fewer than two responses or a nonpositive advantage gap, and report the mean across valid prompts. For an idealized KL mirror step with temperature $\tau > 0$, this statistic equals $\Delta\tilde{A}(s)/\tau$ (Appendix A.1). For a general update, $\tau_{\mathrm{eff}} := \Delta\tilde{A}(s)/\kappa_{\mathrm{eff}}$ is only a pair-specific diagnostic when the denominator is nonzero; it need not be positive. PPO-style surrogate clipping alone does not establish an exact positive-temperature projection or a hard bound on this statistic.

Intuitively, κ_{eff} measures how aggressively RL turns advantage gaps into log-odds gaps after a single update. Hence a larger κ_{eff} implies more sharpening pressure on π_θ. Loosely speaking, although larger κ often indicates a more converging model that may give better performance, an ideal training trajectory should have large κ mainly for genuine learning signals, and preserve room for lower-reward ones that may still contain useful segments to support better overall convergence.

Later, in §4.1, we compare how different advantage constructions shape κ and examine their implications for training dynamics. This motivates reward budgeting, which regulates sharpening pressure through reward allocation rather than explicit entropy control.

Generally speaking, entropy, κ and κ_{eff} differ in granularity and the use of response-quality signals. Entropy measures stepwise output randomness on training data and does not indicate response quality. κ includes $r^{+,-}$ to distinguish good and bad responses, but it is a global metric shared across all steps. κ_{eff} extends κ to the step level, preserving quality distinction while providing finer discrimination.

3.2.1 Sharpness factor controlled studies

Compared with the tested entropy interventions in §3.1, changes to reward scale and KL strength produce clearer differences in finite-horizon training trajectories. We summarize these configurations through κ below; the comparison does not isolate it as a unique causal mediator or determine an asymptotic endpoint.

Specifically, we train Qwen3-8B [29] on DAPO-Math-17K [31] with DRGRPO [16], under three configurations: $(r^+ - r^-, \beta) = (5, 0.001)$, $(2, 0.001)$, and $(1, 0.005)$, yielding decreasing sharpening pressures of $\kappa = 5000, 2000, 200$. To simplify the analysis, we use κ instead of κ_{eff}, since the two serve as proxies for each other. We vary only κ and the related quantities r^+, r^-, and β, while keeping other hyper-parameters fixed.

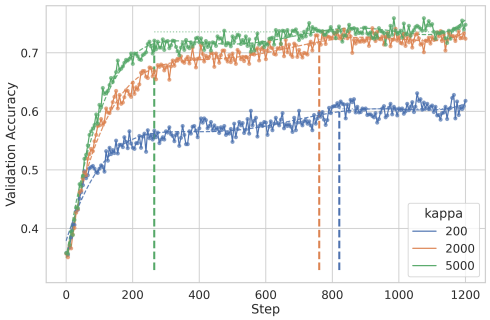


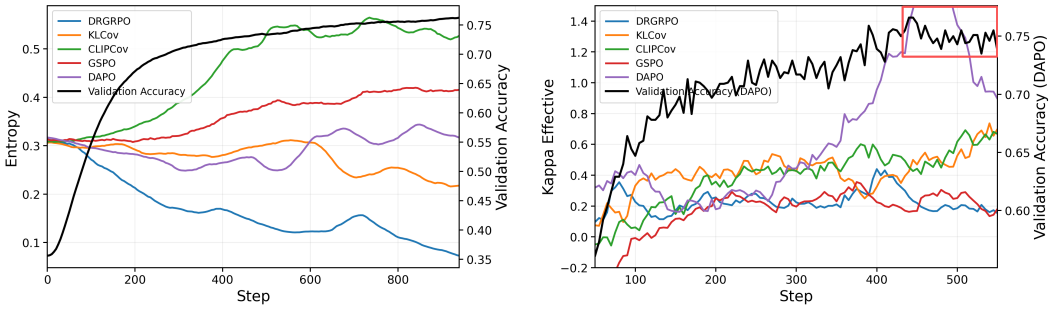
Figure 3: Validation accuracy as training progresses. The three vertical dashed lines mark the convergence step t_{conv}. Overall, smaller κ results in later convergence but also slows training.

For each configuration, we define t_{conv} as the first point where performance plateaus (§B.3). We visualize t_{conv} by vertical lines. In Figure 3, larger κ indeed exhibits markedly earlier t_{conv}, but smaller κ also yields lower final performance. This aligns with §3.2, where κ (and κ_{eff}) are derived from sharpening pressure. Making κ globally small can slow learning substantially *without* guaranteeing higher final accuracy; see more discussion in §4.1.

Under the same set-up, we provide a complementary conditional association analysis in C, showing that entropy is much less predictive of future accuracy once variation in κ is controlled for.

3.3 The averaging effect

When plotting entropy and accuracy, "entropy collapse" appears striking: accuracy plateaus as entropy drops. However, this observation is usually based on heavily averaged curves, such as 11 runs in [3] and 12 runs in our Figure 1. Although averaging reduces noise, it also introduces a subtle **statistical risk**: by smoothing away run-level heterogeneity, it can make weak or inconsistent trends appear visually robust. Therefore, averaged plots and individual runs can show markedly different trends. This distinction is particularly important for causal interpretation. The relationship between a strongly tied metric and accuracy should remain visible within individual runs, not only after averaging.



(a) Entropy over training steps.

(b) κ_{eff} over training steps.

Figure 4: Training dynamics for DRGRPO, KLCov, CLIPCov, GSPO, and DAPO. The left axes show entropy in (a) and κ_{eff} in (b); the right axes show averaged validation accuracy in (a) and DAPO validation accuracy in (b).

Figure 4 shows substantial between-method variation in the diagnostic trajectories. These plots provide descriptive comparisons and do not establish a general within-run relationship or a causal connection between entropy, effective sharpness, and accuracy.

4 Method

We propose a plug-and-play advantage construction method, **reward budgeting**, to regulate RL training dynamics. Despite its simplicity, reward budgeting has a principled theoretical effect. First, under the conditions below, its unclipped score-function estimator follows a finite-G concave objective. Second, it yields a κ trajectory that is different from most existing RL methods which may better allocate training focus. Its empirical gains are further verified in §5.

Reward budgeting changes how advantages are constructed and can be used with a fixed-reference KL penalty. Section 4.2 discusses the reference-free variant, which we evaluate separately in the Qwen3-8B GRPO ablation.

4.1 Fixed per-prompt reward budgets

Algorithm 1 Advantage Construction in Fixed Per-Prompt Reward Budgets

1. Initialize the policy π_θ and set the reference policy $\pi_{\text{ref}} \leftarrow \pi_\theta$; initialize training data $\mathcal{D}$ and the number of rollouts per prompt G.
2. For each training step $t = 1, 2, \ldots$:
   (a) Sample B prompts $\{p_i\}_{i=1}^{B}$ from $\mathcal{D}$.
   (b) Generate G rollouts $\{o_i^{(j)}\}_{j=1}^{G} \sim \pi_\theta$ for each p_i.
   (c) Assign raw rewards $r_i^{(j)}$ to each $o_i^{(j)}$.
   (d) Compute per-prompt minimum $r_i^- := \min_{1 \le j \le G} r_i^{(j)}$.
   (e) Subtract the minimum $g_i^{(j)} := r_i^{(j)} - r_i^-$.
   (f) If a prompt p_i has $g_i^{(j)} = 0$ for all $j \in \{1, \ldots, G\}$, skip the prompt; otherwise:
      i. Redesign rewards $R_i^{(j)} := g_i^{(j)} / \sum_{k=1}^{G} g_i^{(k)}$.
      ii. Compute $A_i^{(j)} := R_i^{(j)} - \frac{1}{G} \sum_j R_i^{(j)}$.

Reward budgeting constructs advantages as follows: for each prompt p, let π_θ sample G rollouts $\{o_i\}_{i=1}^{G}$ with raw rewards $\{r_i\}_{i=1}^{G}$. It first removes a per-group minimum $r^- := \min_i r_i$ and keeps only the positive "gaps" $g_i := r_i - r^-$. If all $g_i = 0$, it skips the update. Otherwise it redistributes a unit total reward across the rollouts, proportional to their gaps: $R(o_i) := g_i / \sum_{j=1}^{G} g_j$ so that $\sum_i R(o_i) = 1$. Finally, it defines the advantage as $A_i := R(o_i) - \frac{1}{G} \sum_i R(o_i)$ to center the normalized rewards within the group; see Algorithm 1. The objective analysis below retains this centering and the group-skipping rule.

Global optimization objectives. To interpret fixed reward budgets, we first compare their idealized score-function update with the standard expected-reward policy gradient [23], whose objective is $J_{\text{lin}} = \mathbb{E}_p \, \mathbb{E}_{y \sim \pi_\theta(\cdot|p)}[R(y)]$. For binary rewards, a sampled group of G rollouts has total raw reward k when k rollouts are correct. Reward budgeting instead assigns a fixed total redistributed reward of 1 to each mixed group. All-correct and all-incorrect groups are skipped because their reward gaps are zero. Thus additional correct rollouts within a mixed group share the same budget; the exact calculation below shows how this produces diminishing returns as success probability increases.

For binary rewards and independent on-policy rollouts, let $p_{\text{correct}} := p_\theta(\text{correct} \mid p)$. Appendix A.3 shows that the expected centered score-function estimator of Algorithm 1, before clipping and other implementation-specific modifications, is the gradient of

$$J_{G-1} = \mathbb{E}_{p \sim \mathcal{D}}\big[f_{G-1}(p_{\text{correct}})\big], \quad f_{G-1}(x) = \sum_{m=1}^{G-1} \frac{1 - (1-x)^m}{m}. \tag{4}$$

This objective is approximately linear up to scale near zero and gives diminishing returns as success probability grows. The result is exact for this estimator at $G = 8$; only its logarithmic limit requires large G.

For general rewards, Appendix A.5 gives sufficient conditions under which the actual gap-normalized estimator has the limiting expected direction associated with

$$J_{\text{gap}} = \mathbb{E}_{p \sim \mathcal{D}} \left[\log \left(\mathbb{E}_{y \sim \pi_\theta}[R(y) \mid p] - a(p) \right) \right], \tag{5}$$

where $a(p)$ is the finite, locally policy-independent essential infimum of the reward distribution. This is an asymptotic interpretation, not an exact finite-G objective for arbitrary graded rewards.

The $O(1/G)$ mean-squared error and variance bounds in Appendix A.4 concern a separate idealized, non-shifted estimator under stronger assumptions; the resulting bias bound is $O(G^{-1/2})$. They neither establish a rate for the actual gap-normalized estimator nor guarantee approximation accuracy at $G = 8$. Clipping, importance weighting, and token-level optimization further limit equivalence to the evaluated implementations.

Sharpness factor. Although reward budgeting applies only a fixed per-prompt reward re-allocation, it turns out to *implicitly* regulate κ in a way that differs from existing RL methods. Specifically, we consider three representative forms of advantage construction:

- DRGRPO uses $A = r - \bar{r}$, giving $\kappa_{\mathrm{DR}} = 1/\beta$;
- GRPO, GSPO, DAPO, ClipCov and KLCov use $A = (r - \bar{r})/\sigma$, giving $\kappa_{\mathrm{GRPO}} = \frac{1}{\beta\sigma}$;
- Reward budgeting gives $\kappa_{RB} = \frac{1}{\beta k}$.

These κ values show how different methods pose sharpening pressure. First, DRGRPO applies uniform sharpening pressure, which is stable but may risk under-convergence. Second, GRPO-style advantages maximize κ at extreme $k \in \{1, G-1\}$, so the strongest sharpening occurs when a batch is almost all correct or almost all incorrect. While this can speed convergence, when a batch is almost all correct, such sharpening is less necessary, because the model is already near perfect. Finally, reward budgeting gives $\kappa = \frac{1}{k\beta}$, assigning stronger sharpening to harder prompts. Thus, "easy" prompts do not keep generating increasingly sharp updates, reducing the incentive to overfit to cheaply repeatable templates.

Table 1: Correct/incorrect samples' advantages in 2 decimal places across different constructions: Reward Budget, DrGRPO and GRPO (most RL methods fall into this type). G is the group size and k is the number of correct rollouts.

	Reward Budget			**DrGRPO**			**GRPO (Most)**		
k/G	$+$	$-$	κ	$+$	$-$	κ	$+$	$-$	κ
1/8	0.88	-0.13	1000	0.88	-0.13	1000	2.65	-0.38	3024
2/8	0.38	-0.13	500	0.75	-0.25	1000	1.73	-0.58	2309
3/8	0.21	-0.13	333	0.63	-0.38	1000	1.29	-0.78	2066
4/8	0.13	-0.13	250	0.50	-0.50	1000	1.00	-1.00	2000
5/8	0.08	-0.13	200	0.38	-0.63	1000	0.78	-1.29	2066
6/8	0.04	-0.13	167	0.25	-0.75	1000	0.58	-1.73	2309
7/8	0.02	-0.13	143	0.13	-0.88	1000	0.38	-2.65	3024
κ form	$1/\beta k$			$1/\beta$			$1/\beta\sigma$		

This sharpness view complements the scoped objective analysis: reward budgeting adjusts the advantage gaps governing idealized sharpening. This provides a mechanism consistent with the observed gains, while the diagnostics do not isolate the causal contribution of each implementation component.

Stepwise KL. Fixed reward budgeting also permits a local surrogate interpretation. For a fixed non-constant sampled group and detached budget weights, consider $\nabla_\theta \sum_{i=1}^{G} R(o_i) \log \pi_\theta(o_i \mid p)$. The nonnegative weights sum to one and define an empirical target distribution, combining weights for duplicate responses. Maximizing this fixed-sample weighted log-likelihood minimizes the KL divergence from that empirical target to the policy, up to a constant. This surrogate omits group-relative centering and clipping; it is not the full implemented objective.

4.2 The prior never fades

Usually π_{ref} is introduced to promote stability by penalizing the current policy for drifting too far, but it in fact plays a deeper role: $\pi^* \propto \pi_{\text{ref}} \exp(\frac{1}{\beta} A)$ permanently depends on π_{ref}, so π_{ref} shapes what the policy is trained to approximate, and its impact never vanishes no matter how long we train.

The reference-free variant removes the KL penalty while retaining importance-ratio clipping. Removing the fixed-reference penalty changes the unconstrained target to reward maximization, whose optima need not be deterministic when rewards tie. Clipping modifies the incentive for large sampled importance-ratio changes around the previous policy; it does not impose a hard global trust region or guarantee a bound on κ_{eff}. The empirical effects of removing KL and adding reward budgeting are distinguished in the ablation study.

Table 2: Benchmark scores (%) from eight responses per prompt at temperature 0.7: mean accuracy for Qwen3-8B and pass@8 for Llama3.2-3B-Instruct. Shaded rows show each baseline's percentage-point change with reward budgeting (RB); GRPO retains reference-policy KL.

Method	AMC23	AIME24	AIME25	MATH500	Olympiad (Physics)	Average
Qwen3-8B						
GRPO	82.50	26.67	43.33	95.60	51.27	59.87
+ RB	+10.00	+10.00	+10.00	+1.20	+2.12	+6.66
DrGRPO	82.50	30.00	43.33	97.20	47.46	60.10
+ RB	+5.00	+6.67	+3.33	-0.20	+1.27	+3.21
GSPO	90.00	26.67	36.67	96.80	47.04	59.44
+ RB	+5.00	+6.67	+6.67	+1.00	+1.69	+4.21
DAPO	87.50	36.67	40.00	96.40	49.15	61.94
+ RB	+7.50	+6.67	+6.67	+1.60	+0.85	+4.66
ClipCov	85.00	26.67	33.33	96.60	48.31	57.98
+ RB	+2.50	+10.00	+6.67	+1.40	+1.27	+4.37
KLCov	82.50	33.33	43.33	96.80	49.58	61.11
+ RB	+5.00	+10.00	+6.67	+0.80	+2.12	+4.92
Llama3.2-3B-Instruct						
GRPO	7.50	3.33	0.00	67.60	20.31	19.75
+ RB	+12.50	+0.00	+3.33	+3.20	+1.53	+4.11
DrGRPO	30.00	10.00	0.00	68.60	14.94	24.71
+ RB	+12.50	+6.67	+6.67	+2.00	+1.15	+5.80
GSPO	22.50	3.33	0.00	72.20	14.83	22.57
+ RB	+7.50	+3.33	+3.33	+5.00	+1.27	+4.01
DAPO	17.50	6.67	3.33	70.40	21.84	23.95
+ RB	+7.50	+3.33	+0.00	+2.20	+1.47	+2.90
ClipCov	22.50	10.00	0.00	69.80	13.98	23.26
+ RB	+12.50	+3.33	+3.33	+3.40	+1.27	+4.77
KLCov	30.00	10.00	0.00	71.60	13.98	25.12
+ RB	+10.0	+6.67	+3.33	+2.40	+1.69	+4.82

5 Experiment

5.1 Setup

Data. We train on DAPO-Math-17K [31] and evaluate on five benchmarks: AMC23 [1], AIME24 [9], AIME25 [34], MATH500 [8], and OlympiadBench (Physics) [6].

Models. We train (1) Llama3.2-3B-Instruct [17] and (2) Qwen3-8B [29].

Hyper-parameters. We sample eight rollouts per prompt and use 32 prompts per update. We train each method until its validation accuracy reaches a plateau. During evaluation, we set the maximum response length to 8192 tokens and generate 8 responses per prompt at temperature$=0.7$. We report their average accuracy for Qwen3-8B and pass@8 for Llama3.2-3B-Instruct (see detailed settings in §B).

Baselines. We apply our method to six RL methods in two types: (1) Four Representative RL Methods: GRPO [21], DrGRPO [16], GSPO [35], and DAPO [31]; (2) Two Entropy-Based RL Methods: ClipCov and KLCov [3], which are entropy-based variants of DAPO.

5.2 Key insights

More robust gains than entropy control. Table 2 includes two remedies: entropy (ClipCov and KLCov as entropy-based variants of DAPO) and our method. However, within each dataset, results of entropy-control methods are mixed. On average, on Qwen3-8B, DAPO even outperforms its variants (DAPO 61.94 vs. ClipCov 57.98 vs. KLCov 61.11); on Llama3.2-3B, KLCov marginally outperforms DAPO (25.12 vs. 23.95) while ClipCov slightly under-performs (23.26 vs. 23.95). In contrast, our method consistently improves in almost all cases.

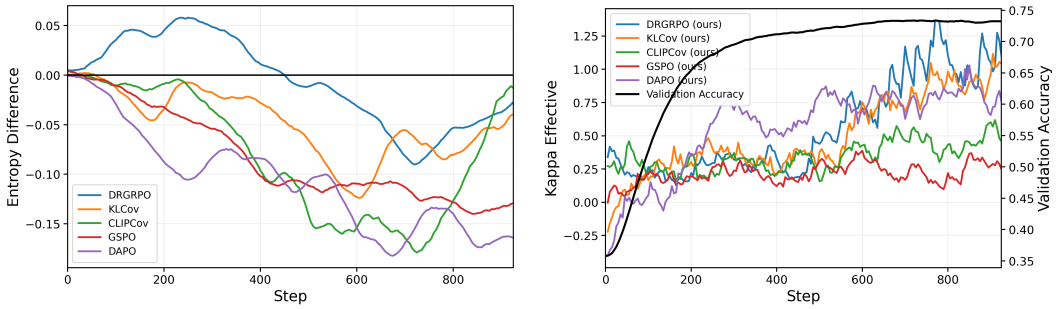
Impact of advantage construction on κ-dynamics. Apart from reward budgeting, the RL baselines in Table 2 construct advantages in mainly two ways: DRGRPO uses $A = r - \bar{r}$, while others use

Table 3: Qwen3-8B GRPO ablation of reference-policy KL and reward budgeting (RB); the first two rows give the absolute scores in Table 2.

KL	RB	AMC23	AIME24	AIME25	MATH500	Olympiad (Physics)	Average
✓	×	82.50	26.67	43.33	95.60	51.27	59.87
✓	✓	92.50	36.67	53.33	96.80	53.39	**66.53**
×	×	85.00	30.00	40.00	96.00	50.35	60.27
×	✓	95.00	36.67	46.67	97.00	52.89	65.65

$A = (r - \bar{r})/\sigma$ (in fact, DAPO and its variants also includes length-based reward, but its magnitude is typically much smaller than r, thus we omit this for simplicity). These different advantage constructions then induce different κ-trajectories (§4.1). Specifically, DRGRPO tends to exhibit slightly lower κ, thus gentler sharpening pressure. Comparing DRGRPO to the others, we observe that its performance remains in the top tier. This agrees with our earlier discussion: well-regulated κ can deliver better final convergence, though our method continues to improve even in this case.

5.3 Training dynamics



(a) Entropy changes after applying our method

(b) κ_{eff} vs validation accuracy

Figure 5: Reward budgeting on Qwen3-8B: (a) training-entropy differences from each baseline, mostly below zero; (b) effective sharpness and mean validation accuracy across methods (black curve). These descriptive trajectories do not establish the cause of the accuracy gains.

From Figure 5a, after applying our method, entropy generally decreases, since most curves remain below the black zero line. This indicates that higher entropy is not necessary for improving final performance, and that our method does not achieve its gains by implicitly boosting entropy. Comparing Figures 5b and 4b, our method also regulates the κ_{eff} trajectories with noticeably smaller oscillations, which is consistent with our discussion in §4.1. These trajectories accompany accuracy gains without a sustained increase in entropy.

5.4 Ablation

The Qwen3-8B GRPO comparison in Table 2 retains the reference-policy KL penalty both with and without reward budgeting. To assess how this penalty affects the result, Table 3 crosses reward budgeting with KL retention or removal.

Removing KL alone changes the average score from 59.87% to 60.27%, with mixed benchmark-level effects (Table 3). Reward budgeting improves the average by $+6.66$ percentage points with KL and $+5.38$ without KL, improving every evaluated benchmark in both settings. Thus, the gains are primarily attributable to reward budgeting rather than KL removal.

6 Conclusion

We study *premature convergence* in RL and suggest that aggregate entropy may not fully characterize training dynamics. Instead, RL's implicit target π^* reveals a built-in sharpening pressure, which is captured with the sharpness factor κ (and κ_{eff}). Motivated by this, we propose reward-budgeting to improve final performance without explicit entropy control. Appendix D discusses limitations of these findings.

Acknowledgments and Disclosure of Funding

This work was supported by Alibaba; the authors declare no other competing interests.

References

[1] AI-MO Team. AIMO Validation Set – AMC Subset. https://huggingface.co/datasets/AI-MO/aimo-validation-amc, 2024. Validation set of AMC 2022–2024 problems; we use the AMC 2023 split (AMC23).

[2] Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective, 2025.

[3] Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models, 2025.

[4] Yaru Hao, Li Dong, Xun Wu, Shaohan Huang, Zewen Chi, and Furu Wei. On-policy rl with optimal reward baseline, 2025.

[5] Zhezheng Hao, Hong Wang, Haoyang Liu, Jian Luo, Jiarui Yu, Hande Dong, Qiang Lin, Can Wang, and Jiawei Chen. Rethinking entropy interventions in RLVR: An entropy change perspective. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31105–31133, San Diego, California, United States, July 2026. Association for Computational Linguistics.

[6] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[7] Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. Skywork open reasoner 1 technical report, 2025.

[8] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

[9] Hugging Face H4. AIME24: AIME 2024 benchmark. https://huggingface.co/datasets/HuggingFaceH4/aime_2024, 2024. 30 problems from the 2024 AIME I and II exams.

[10] Yuxian Jiang, Yafu Li, Guanxu Chen, Dongrui Liu, Yu Cheng, and Jing Shao. Rethinking entropy regularization in large reasoning models, 2025.

[11] Renren Jin, Pengzhi Gao, Yuqi Ren, Zhuowen Han, Tongxuan Zhang, Wuwei Huang, Wei Liu, Jian Luan, and Deyi Xiong. Revisiting entropy in reinforcement learning for large reasoning models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Findings of the Association for Computational Linguistics: ACL 2026*, pages 25300–25322, San Diego, California, United States, July 2026. Association for Computational Linguistics.

[12] Devvrit Khatri, Lovish Madaan, Rishabh Tiwari, Rachit Bansal, Venkata Sai Surya Subramanyam Duvvuri, Manzil Zaheer, Inderjit Dhillon, David Brandfonbrener, and Rishabh Agarwal. The art of scaling reinforcement learning compute for llms. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust, editors, *International Conference on Learning Representations*, volume 2026, pages 72438–72467, 2026.

[13] Shiye Lei, Zhihao Cheng, Kai Jia, and Dacheng Tao. Revisiting llm reasoning via information bottleneck, 2025.

[14] Cheuk Ting Li and Farzan Farnia. Mode-seeking divergences: Theory and applications to gans. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 8321–8350. PMLR, 25–27 Apr 2023.

[15] Qingbin Li, Rongkun Xue, Jie Wang, Ming Zhou, Zhi Li, Xiaofeng Ji, Yongqi Wang, Miao Liu, Zheming Yang, Minghui Qiu, and Jing Yang. Cure: Critical-token-guided re-concatenation for entropy-collapse prevention, 2025.

[16] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Conference on Language Modeling (COLM)*, 2025.

[17] Meta. Llama-3.2-3b-instruct. Hugging Face model card, 2024.

[18] MiniMax, :, Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, Chengjun Xiao, Chengyu Du, Chi Zhang, Chu Qiao, Chunhao Zhang, Chunhui Du, Congchao Guo, Da Chen, Deming Ding, Dianjun Sun, Dong Li, Enwei Jiao, Haigang Zhou, Haimo Zhang, Han Ding, Haohai Sun, Haoyu Feng, Huaiguang Cai, Haichao Zhu, Jian Sun, Jiaqi Zhuang, Jiaren Cai, Jiayuan Song, Jin Zhu, Jingyang Li, Jinhao Tian, Jinli Liu, Junhao Xu, Junjie Yan, Junteng Liu, Junxian He, Kaiyi Feng, Ke Yang, Kecheng Xiao, Le Han, Leyang Wang, Lianfei Yu, Liheng Feng, Lin Li, Lin Zheng, Linge Du, Lingyu Yang, Lunbin Zeng, Minghui Yu, Mingliang Tao, Mingyuan Chi, Mozhi Zhang, Mujie Lin, Nan Hu, Nongyu Di, Peng Gao, Pengfei Li, Pengyu Zhao, Qibing Ren, Qidi Xu, Qile Li, Qin Wang, Rong Tian, Ruitao Leng, Shaoxiang Chen, Shaoyu Chen, Shengmin Shi, Shitong Weng, Shuchang Guan, Shuqi Yu, Sichen Li, Songquan Zhu, Tengfei Li, Tianchi Cai, Tianrun Liang, Weiyu Cheng, Weize Kong, Wenkai Li, Xiancai Chen, Xiangjun Song, Xiao Luo, Xiao Su, Xiaobo Li, Xiaodong Han, Xinzhu Hou, Xuan Lu, Xun Zou, Xuyang Shen, Yan Gong, Yan Ma, Yang Wang, Yiqi Shi, Yiran Zhong, Yonghong Duan, Yongxiang Fu, Yongyi Hu, Yu Gao, Yuanxiang Fan, Yufeng Yang, Yuhao Li, Yulin Hu, Yunan Huang, Yunji Li, Yunzhi Xu, Yuxin Mao, Yuxuan Shi, Yuze Wenren, Zehan Li, Zelin Li, Zhanxu Tian, Zhengmao Zhu, Zhenhua Fan, Zhenzhen Wu, Zhichao Xu, Zhihang Yu, Zhiheng Lyu, Zhuo Jiang, Zibo Gao, Zijia Wu, Zijian Song, and Zijun Sun. Minimax-m1: Scaling test-time compute efficiently with lightning attention, 2025.

[19] Jaesung R. Park, Junsu Kim, Gyeongman Kim, Jinyoung Jo, Sean Choi, Jaewoong Cho, and Ernest K. Ryu. Clip-low increases entropy and clip-high decreases entropy in reinforcement learning of large language models, 2025.

[20] Aleksei Petrenko, Ben Lipkin, Kevin Chen, Erik Wijmans, Marco Cusumano-Towner, Raja Giryes, and Philipp Krähenbühl. Entropy-preserving reinforcement learning, 2026.

[21] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

[22] Zhenpeng Su, Leiyu Pan, Minxuan Lv, Yuntao Li, Wenping Hu, Fuzheng Zhang, Kun Gai, and Guorui Zhou. CE-GPPO: Coordinating entropy via gradient-preserving clipping policy optimization in reinforcement learning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 37747–37760, San Diego, California, United States, July 2026. Association for Computational Linguistics.

[23] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999.

[24] Hongze Tan, Zihan Wang, Jianfei Pan, Jinghao Lin, Hao Wang, Yifan Wu, Tao Chen, Zhihang Zheng, Zhihao Tang, and Haihua Yang. GTPO and GRPO-s: Token and sequence-level reward shaping with policy entropy. In Tong Zhang, Miroslav Dudik, Martin Jaggi, Alekh Agarwal, Sharon Li, Dale Schuurmans, Jerry Zhu, Felix Berkenkamp, Hanze Dong, and Alberto Bietti, editors, *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 118609–118630. PMLR, 06–11 Jul 2026.

[25] Chen Wang, Zhaochun Li, Jionghao Bai, Yuzhi Zhang, Shisheng Cui, Zhou Zhao, and Yue Wang. Arbitrary entropy policy optimization breaks the exploration bottleneck of reinforcement learning. *arXiv preprint arXiv:2510.08141*, 2025.

[26] Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 115452–115486. Curran Associates, Inc., 2025.

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

[28] Wujiang Xu, Wentian Zhao, Zhenting Wang, Yu-Jhe Li, Can Jin, Mingyu Jin, Kai Mei, Kun Wan, and Dimitris N. Metaxas. Epo: Entropy-regularized policy optimization for llm agents reinforcement learning, 2026.

[29] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025.

[30] Kai Yang, Xin Xu, Yangkun Chen, Weijie Liu, Jiafei Lyu, Zichuan Lin, Deheng Ye, and Saiyong Yang. Entropic: Towards stable long-term training of llms via entropy stabilization with proportional-integral control, 2026.

[31] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025.

[32] Xiaoyun Zhang, Xiaojian Yuan, Di Huang, Wang You, Chen Hu, Jingqing Ruan, Kejiang Chen, and Xing Hu. Revisiting entropy regularization: Adaptive coefficient unlocks its potential for LLM reinforcement learning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Findings of the Association for Computational Linguistics: ACL 2026*, pages 18005–18020, San Diego, California, United States, July 2026. Association for Computational Linguistics.

[33] Yanzhi Zhang, Zhaoxi Zhang, Haoxiang Guan, Yilin Cheng, Yitong Duan, Chen Wang, Yue Wang, Shuxin Zheng, and Jiyan He. No free lunch: Rethinking internal feedback for llm reasoning, 2025.

[34] Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

[35] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025.

A Proofs

A.1 The effective kappa

Lemma A.1 (Effective sharpness for a KL mirror step)**.** *Fix a state s, a finite action space, a full-support policy ρ, detached advantages $\tilde{A}(s,a)$, and $\tau > 0$. For the KL-regularized projection below and the log-odds definition in Eq. 3,*

$$\kappa_{\text{eff}}(s; a^+, a^-) = \frac{\tilde{A}(s,a^+) - \tilde{A}(s,a^-)}{\tau}.$$

This identity describes the idealized projection, not an exact representation of arbitrary clipped policy updates.

Proof. Fix a prompt/state s and write $\pi(a) \equiv \pi(a \mid s)$, $\rho(a) \equiv \rho(a \mid s)$, and $\tilde{A}(a) \equiv \tilde{A}(s,a)$. Consider the KL-regularized projection (mirror descent) update

$$\pi_{\text{new}} \in \arg\max_{\pi} \left\{ \sum_a \pi(a)\, \tilde{A}(a) - \tau\, D_{\text{KL}}(\pi \| \rho) \right\} \tag{6}$$

such that

$$\sum_a \pi(a) = 1, D_{\text{KL}}(\pi\|\rho) := \sum_a \pi(a) \log \frac{\pi(a)}{\rho(a)}. \tag{7}$$

Introduce the Lagrangian

$$\mathcal{L}(\pi, \lambda) = \sum_a \pi(a)\tilde{A}(a) - \tau \sum_a \pi(a) \log \frac{\pi(a)}{\rho(a)} + \lambda\Big(\sum_a \pi(a) - 1\Big). \tag{8}$$

Taking derivatives and setting them to zero, for each a we have

$$\begin{aligned} 0 = \frac{\partial \mathcal{L}}{\partial \pi(a)} &= \tilde{A}(a) - \tau\Big(\log \frac{\pi(a)}{\rho(a)} + 1\Big) + \lambda, \\ \log \frac{\pi(a)}{\rho(a)} &= \frac{\tilde{A}(a)}{\tau} + \frac{\lambda - \tau}{\tau}, \\ \pi(a) &= \rho(a) \exp\Big(\frac{\tilde{A}(a)}{\tau}\Big) \cdot \exp\Big(\frac{\lambda - \tau}{\tau}\Big). \end{aligned} \tag{9}$$

Let $C := \exp((\lambda - \tau)/\tau)$, which does not depend on a. Enforcing normalization yields

$$\begin{aligned} 1 = \sum_a \pi(a) &= C \sum_a \rho(a) \exp\Big(\frac{\tilde{A}(a)}{\tau}\Big) \\ \implies \quad C &= \left(\sum_{a'} \rho(a') \exp\Big(\frac{\tilde{A}(a')}{\tau}\Big) \right)^{-1}. \end{aligned} \tag{10}$$

Therefore,

$$\pi_{\text{new}}(a \mid s) = \frac{\rho(a \mid s) \exp(\tilde{A}(s,a)/\tau)}{\sum_{a'} \rho(a' \mid s) \exp(\tilde{A}(s,a')/\tau)}. \tag{11}$$

For any two actions/outputs a^+ and a^-, the normalizer cancels:

$$\frac{\pi_{\text{new}}(a^+ \mid s)}{\pi_{\text{new}}(a^- \mid s)} = \frac{\rho(a^+ \mid s)}{\rho(a^- \mid s)} \exp\Big(\frac{\tilde{A}(s,a^+) - \tilde{A}(s,a^-)}{\tau}\Big), \tag{12}$$

$$\log \frac{\pi_{\text{new}}(a^+ \mid s)}{\pi_{\text{new}}(a^- \mid s)} - \log \frac{\rho(a^+ \mid s)}{\rho(a^- \mid s)} = \frac{\tilde{A}(s,a^+) - \tilde{A}(s,a^-)}{\tau}. \tag{13}$$

Defining $\Delta\tilde{A}(s) := \tilde{A}(s,a^+) - \tilde{A}(s,a^-)$ and

$$\kappa_{\text{eff}}(s; a^+, a^-) := \log \frac{\pi_{\text{new}}(a^+ \mid s)}{\pi_{\text{new}}(a^- \mid s)} - \log \frac{\rho(a^+ \mid s)}{\rho(a^- \mid s)}, \tag{14}$$

we obtain the claimed identity

$$\kappa_{\text{eff}}(s; a^+, a^-) = \frac{\Delta\tilde{A}(s)}{\tau} \quad (\text{and } \tau_{\text{eff}} = \tau \text{ when } \Delta\tilde{A}(s) \neq 0). \tag{15}$$

□

A.2 Scope of the objective analysis

Algorithm 1 uses a gap-normalized reward-budget estimator: it first subtracts the within-group minimum reward, normalizes the remaining positive gaps to a fixed per-prompt budget, and skips groups whose gaps are all zero. This differs from the simpler reward-normalized estimator that directly uses $R_i / \sum_j R_j$.

We therefore separate the objective analysis into three cases. First, in Appendix A.3, we derive the exact finite-G objective induced by Algorithm 1 in the binary-reward setting. This is the setting most closely aligned with our main RLVR experiments. Second, in Appendix A.4, we analyze an idealized non-shifted reward-normalized estimator, which gives the clean $\nabla \log \mathbb{E}[R]$ interpretation. This result should be understood as an intuition for why fixed reward budgets induce diminishing returns, rather than as the exact finite-G objective of Algorithm 1. Third, in Appendix A.5, we discuss the actual gap-normalized estimator under general rewards and show that, under additional support assumptions, it asymptotically corresponds to a shifted log-mean objective.

A.3 Exact objective of Algorithm 1 under binary rewards

We first analyze the estimator exactly induced by Algorithm 1 in the binary reward setting. This differs slightly from the simpler c_i/k estimator, because Algorithm 1 subtracts the group minimum reward and skips groups whose post-shift gaps are all zero.

Assumption A.2 (Binary reward setting)**.** For a prompt x, sample G i.i.d. rollouts $y_1, \ldots, y_G \sim \pi_\theta(\cdot|x)$. Let $c(y) \in \{0, 1\}$ denote correctness, $c_i := c(y_i)$, and

$$k := \sum_{i=1}^{G} c_i, \qquad p_\theta(x) := \Pr_{y \sim \pi_\theta(\cdot|x)}(c(y) = 1).$$

Assume $G \geq 2$, $0 < p_\theta(x) < 1$, a correctness rule with no explicit θ dependence, locally parameter-independent policy support, and the integrability and differentiation conditions needed for the score-function identities below.

Under Algorithm 1, there are three cases:

$$k = 0 : \text{ all rollouts are incorrect and the group is skipped,}$$

$$1 \leq k \leq G - 1 : \text{ the group is mixed and contributes an update,}$$

$$k = G : \text{ all rollouts are correct and the group is skipped.}$$

The last case is important: after subtracting the group minimum, all gaps become zero, so the group contains no within-group preference contrast.

For $1 \leq k \leq G - 1$, the shifted gaps are $g_i = c_i$, the normalized budgeted rewards are $R_i = c_i/k$, and Algorithm 1 uses the group-relative advantage

$$A_i = \frac{c_i}{k} - \frac{1}{G}.$$

Thus the exact estimator is

$$\hat{g}_{\mathrm{RB}}(x) = \mathbf{1}\{1 \leq k \leq G - 1\} \sum_{i=1}^{G} \left(\frac{c_i}{k} - \frac{1}{G} \right) \nabla_\theta \log \pi_\theta(y_i|x).$$

Lemma A.3 (Exact binary objective of Algorithm 1)**.** *For binary rewards, the expected update of Algorithm 1 satisfies*

$$\mathbb{E}[\hat{g}_{\mathrm{RB}}(x)] = \nabla_\theta f_{G-1}(p_\theta(x)),$$

where

$$f_H(p) = \sum_{m=1}^{H} \frac{1 - (1-p)^m}{m}, \qquad H = G - 1.$$

Moreover, f_H is concave in p.

Proof. Let

$$\mu_+ := \mathbb{E}[\nabla_\theta \log \pi_\theta(y|x) \mid c(y) = 1], \qquad \mu_- := \mathbb{E}[\nabla_\theta \log \pi_\theta(y|x) \mid c(y) = 0].$$

By the score-function identity,

$$\mu_+ = \nabla_\theta \log p_\theta(x), \qquad \mu_- = \nabla_\theta \log(1 - p_\theta(x)).$$

Conditioned on $k \in \{1, \ldots, G-1\}$, there are k correct samples and $G-k$ incorrect samples. Therefore,

$$\mathbb{E}[\hat{g}_{\mathrm{RB}}(x) \mid k] = k\left(\frac{1}{k} - \frac{1}{G}\right)\mu_+ + (G-k)\left(-\frac{1}{G}\right)\mu_-.$$

This simplifies to

$$\mathbb{E}[\hat{g}_{\mathrm{RB}}(x) \mid k] = \frac{G-k}{G}(\mu_+ - \mu_-).$$

Since $k \sim \mathrm{Binomial}(G, p)$, we have

$$\mathbb{E}\left[\frac{G-k}{G}\mathbf{1}\{1 \le k \le G-1\}\right] = (1-p) - (1-p)^G.$$

Hence

$$\mathbb{E}[\hat{g}_{\mathrm{RB}}(x)] = \big((1-p) - (1-p)^G\big)(\mu_+ - \mu_-).$$

Using

$$\mu_+ - \mu_- = \nabla_\theta \log p - \nabla_\theta \log(1-p) = \frac{\nabla_\theta p}{p(1-p)},$$

we obtain

$$\mathbb{E}[\hat{g}_{\mathrm{RB}}(x)] = \frac{1 - (1-p)^{G-1}}{p}\nabla_\theta p.$$

Therefore this is the gradient of a scalar objective $f_{G-1}(p)$ satisfying

$$f'_{G-1}(p) = \frac{1 - (1-p)^{G-1}}{p} = \sum_{m=0}^{G-2}(1-p)^m.$$

Integrating gives

$$f_{G-1}(p) = \sum_{m=1}^{G-1}\frac{1 - (1-p)^m}{m}$$

up to an additive constant.

Finally,

$$f''_{G-1}(p) = -\sum_{m=1}^{G-2} m(1-p)^{m-1} \le 0,$$

so f_{G-1} is concave. □

For fixed $p \in (0, 1]$, let $H_n = \sum_{m=1}^{n} 1/m$. Then

$$f_n(p) - H_n = -\sum_{m=1}^{n}\frac{(1-p)^m}{m} \longrightarrow \log p.$$

Thus the logarithmic limit is understood up to a parameter-independent additive constant; $f_n(p)$ itself does not have a finite limit. The score-function derivation assumes $0 < p_\theta(x) < 1$ and the usual differentiation-under-expectation conditions.

A.4 Idealized non-shifted reward-normalized estimator

Assumption A.4 (General reward setting)**.** Fix a prompt x. Let $y_1, \dots, y_G \overset{\text{i.i.d.}}{\sim} \pi_\theta(\cdot \mid x)$. Define

$$R_i := R(y_i), \quad S := \sum_{j=1}^{G} R_j, \quad \mu_\theta(x) := \mathbb{E}_{y \sim \pi_\theta(\cdot|x)}[R(y)].$$

$$\phi_i := \nabla_\theta \log \pi_\theta(y_i \mid x) \in \mathbb{R}^d.$$

Assume:

1. There are fixed constants $0 < r_{\text{lo}} \le r_{\text{hi}} < \infty$ such that $r_{\text{lo}} \le R(y) \le r_{\text{hi}}$ almost surely. These are sufficient conditions for this idealized estimator only; zero-valued binary rewards are covered separately by the exact binary analysis.
2. $\mathbb{E}[\|\phi(y)\|^2 \mid x] < \infty$, where $\phi(y) = \nabla_\theta \log \pi_\theta(y \mid x)$.
3. Rewards have no explicit dependence on θ, the policy has parameter-independent support locally, and differentiation may pass under the relevant expectations. Thus $\mathbb{E}[\phi] = 0$ and $\mathbb{E}[R\phi] = \nabla_\theta \mu_\theta(x)$.

All expectations below are conditional on x and we omit the conditioning for brevity.

Asymptotic-G derivation

Define per-sample normalized rewards

$$\tilde{r}_i = \begin{cases} R_i/S, & S > 0, \\ 0, & S = 0, \end{cases}$$

and the VPG/REINFORCE [27, 23] group estimator

$$\begin{aligned} g_{\text{new}}(x) &:= \sum_{i=1}^{G} \tilde{r}_i \, \nabla_\theta \log \pi_\theta(y_i \mid x) \\ &= \mathbf{1}[S > 0] \sum_{i=1}^{G} \frac{R_i}{S} \nabla_\theta \log \pi_\theta(y_i \mid x). \end{aligned}$$

Lemma A.5 (large G objective)**.** *Under the assumptions above,*

$$\lim_{G \to \infty} \mathbb{E}[g_{\text{new}}(x)] = \nabla_\theta \log \mu_\theta(x).$$

Consequently, for large G the implicit objective is

$$J_{\text{new}} \approx \mathbb{E}_x[\log \mu_\theta(x)].$$

Proof. Write $\bar{R}_G = G^{-1} \sum_i R_i$, $\bar{\psi}_G = G^{-1} \sum_i R_i \phi_i$, $\mu = \mathbb{E}[R]$, and $A = \mathbb{E}[R\phi] = \nabla_\theta \mu$. The estimator is $g_{\text{new}} = \bar{\psi}_G / \bar{R}_G$ and $\bar{R}_G \ge r_{\text{lo}} > 0$. Therefore

$$g_{\text{new}} - \frac{A}{\mu} = \frac{(\bar{\psi}_G - A) - (A/\mu)(\bar{R}_G - \mu)}{\bar{R}_G}.$$

Independence and the finite second moments give

$$\mathbb{E}\left\| g_{\text{new}} - \frac{A}{\mu} \right\|^2 \le \frac{2}{r_{\text{lo}}^2 G} \left(\mathbb{E}\|R\phi - A\|^2 + \frac{\|A\|^2}{\mu^2} \operatorname{Var}(R) \right).$$

Hence $g_{\text{new}} \to A/\mu$ in L^2 and therefore in L^1. Taking expectations yields $\mathbb{E}[g_{\text{new}}] \to \nabla_\theta \log \mu$. The lower reward bound controls the random denominator. □

Finite-G approximations

Lemma A.6 (L^2-rate of g_G around $\nabla \log \mu$)**.** *Under Assumption A.4, there exists a finite constant $C(x) > 0$ (depending on the distribution at prompt x but not on G) such that for all sufficiently large G,*

$$\mathbb{E}\big[\|g_G(x) - \nabla_\theta \log \mu_\theta(x)\|^2\big] \le \frac{C(x)}{G}.$$

Proof. Write $g_G = g_{\text{new}}$ for a group of size G. The direct ratio bound in the proof of Lemma A.5 applies for every $G \ge 1$. The claim follows by choosing

$$C(x) = \frac{2}{r_{\text{lo}}^2}\left(\mathbb{E}\|R\phi - A\|^2 + \frac{\|A\|^2}{\mu^2}\operatorname{Var}(R)\right),$$

or any larger positive constant. It is finite under Assumption A.4. □

Corollary A.7 (Bias, variance and MSE of g_G)**.** *Under Assumption A.4, there exists $C(x) < \infty$ such that for all sufficiently large G,*

$$\big\|\mathbb{E}[g_G(x)] - \nabla_\theta \log \mu_\theta(x)\big\| \le \sqrt{\frac{C(x)}{G}}$$

$$\operatorname{Var}\big(g_G(x)\big) := \mathbb{E}\big[\|g_G(x) - \mathbb{E}[g_G(x)]\|^2\big] \;\le\; \frac{C(x)}{G}$$

Proof. The mean-squared error bound is Lemma A.6. The bias bound follows from Cauchy–Schwarz:

$$\big\|\mathbb{E}[g_G(x)] - \nabla_\theta \log \mu_\theta(x)\big\|^2 \le \mathbb{E}\big[\|g_G(x) - \nabla_\theta \log \mu_\theta(x)\|^2\big].$$

For the variance,

$$\begin{aligned}\operatorname{Var}\big(g_G(x)\big) &= \mathbb{E}\big[\|g_G(x) - \mathbb{E}[g_G(x)]\|^2\big] \\ &\le \mathbb{E}\big[\|g_G(x) - \nabla_\theta \log \mu_\theta(x)\|^2\big],\end{aligned}$$

since $\nabla_\theta \log \mu_\theta(x)$ is a fixed vector. The last quantity is bounded by $C(x)/G$ by Lemma A.6. □

A.5 Gap-normalized estimator under general rewards

We now give sufficient conditions for the actual gap-normalized estimator. Fix a prompt x and policy θ, and draw an i.i.d. sequence of rollouts. Write $R_i = R(y_i, x)$, $\phi_i = \nabla_\theta \log \pi_\theta(y_i \mid x)$, $M_G = \min_{i \le G} R_i$, and $Z_G = \sum_i (R_i - M_G)$. Let $a(x)$ be the finite essential infimum of R under π_θ, assumed policy-independent in a neighborhood of θ. Assume $\mu = \mathbb{E}[R] > a(x)$, and

$$\mathbb{E}|R| < \infty, \qquad \mathbb{E}\|\phi\| < \infty, \qquad \mathbb{E}[|R|\|\phi\|] < \infty.$$

Assume also that rewards have no explicit θ dependence, the policy has locally parameter-independent support, and differentiation may pass under the expectation, so that $\mathbb{E}[\phi] = 0$ and $\mathbb{E}[R\phi] = \nabla_\theta \mu$.

For $Z_G > 0$, the centered Algorithm 1 score-function estimator is

$$\hat{g}_{\text{gap}} = \sum_{i=1}^{G}\left(\frac{R_i - M_G}{Z_G} - \frac{1}{G}\right)\phi_i;$$

it is zero when $Z_G = 0$. The centered term cannot in general be discarded by treating the group-dependent skipping rule as an action-independent event. Instead, with $\bar{\phi}_G = G^{-1}\sum_i \phi_i$, rewrite the estimator as

$$\hat{g}_{\text{gap}} = \frac{G^{-1}\sum_i R_i\phi_i - M_G\bar{\phi}_G}{G^{-1}\sum_i R_i - M_G} - \bar{\phi}_G \quad \text{when } Z_G > 0.$$

The sampled minimum converges almost surely to $a(x)$. By the strong law, the numerator converges to $\nabla_\theta \mu$, the denominator converges to $\mu - a(x) > 0$, and $\bar{\phi}_G \to 0$. Thus

$$\hat{g}_{\text{gap}} \xrightarrow{a.s.} \frac{\nabla_\theta \mu}{\mu - a(x)} = \nabla_\theta \log(\mu - a(x)).$$

Convergence of the expected update additionally requires uniform integrability. One sufficient condition is an almost-sure bound $\|\phi\| \le L < \infty$: the nonnegative normalized gaps sum to one, so $\|\hat{g}_{\mathrm{gap}}\| \le 2L$, and dominated convergence then applies. Under these sufficient conditions, the large-group expected direction has a shifted log-mean interpretation. This argument establishes neither an exact finite-G general-reward objective nor a finite-G rate; the behavior of the sampled minimum can depend on the reward distribution near its lower endpoint. When $a(x) = 0$, the limiting expression coincides with the unshifted log-mean.

A.6 RL as policy-to-target KL minimization

A.6.1 Trajectory level

Claim. The regularized RL objective

$$J_{\mathrm{RL}}(\theta) = \mathbb{E}_{\gamma\sim\pi_\theta}\big[R(\gamma)\big] - \beta\, D_{\mathrm{KL}}\big(\pi_\theta \,\|\, \pi_{\mathrm{ref}}\big)$$

is equivalent (up to an additive constant independent of θ) to

$$-\beta\, D_{\mathrm{KL}}\big(\pi_\theta \,\|\, \pi^*_{\mathrm{RL}}\big),$$

where

$$\pi^*_{\mathrm{RL}}(\gamma) \propto \pi_{\mathrm{ref}}(\gamma) \exp\left(\frac{R(\gamma)}{\beta}\right).$$

Proof. Let γ denote a trajectory (rollout) and let $\pi_\theta(\gamma)$ be the trajectory distribution induced by policy π_θ. Consider

$$J_{\mathrm{RL}}(\theta) = \mathbb{E}_{\gamma\sim\pi_\theta}[R(\gamma)] - \beta\, D_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}}).$$

Expanding the KL term,

$$D_{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}}) = \mathbb{E}_{\gamma\sim\pi_\theta}\left[\log \frac{\pi_\theta(\gamma)}{\pi_{\mathrm{ref}}(\gamma)}\right],$$

so

$$J_{\mathrm{RL}}(\theta) = \mathbb{E}_{\gamma\sim\pi_\theta}[R(\gamma) - \beta \log \pi_\theta(\gamma) + \beta \log \pi_{\mathrm{ref}}(\gamma)]\,.$$

Define the (nonparametric) optimal distribution

$$\pi^*_{\mathrm{RL}}(\gamma) = \frac{1}{Z}\, \pi_{\mathrm{ref}}(\gamma) \exp\left(\frac{R(\gamma)}{\beta}\right),$$
$$Z = \sum_\gamma \pi_{\mathrm{ref}}(\gamma) \exp\left(\frac{R(\gamma)}{\beta}\right).$$

Then

$$\log \pi^*_{\mathrm{RL}}(\gamma) = \log \pi_{\mathrm{ref}}(\gamma) + \frac{R(\gamma)}{\beta} - \log Z.$$

Compute the policy-to-target KL:

$$\begin{aligned} &D_{\mathrm{KL}}(\pi_\theta \| \pi^*_{\mathrm{RL}}) \\ &= \mathbb{E}_{\gamma\sim\pi_\theta}[\log \pi_\theta(\gamma) - \log \pi^*_{\mathrm{RL}}(\gamma)] \\ &= \mathbb{E}_{\gamma\sim\pi_\theta}\left[\log \pi_\theta(\gamma) - \log \pi_{\mathrm{ref}}(\gamma) - \frac{R(\gamma)}{\beta} + \log Z\right] \end{aligned}$$

Multiplying by $-\beta$ gives

$$\begin{aligned} -\beta D_{\mathrm{KL}}(\pi_\theta \| \pi^*_{\mathrm{RL}}) &= \mathbb{E}[R(\gamma) - \beta \log \pi_\theta(\gamma) + \beta \log \pi_{\mathrm{ref}}(\gamma)] \\ -\beta \log Z &= J_{\mathrm{RL}}(\theta) - \beta \log Z. \end{aligned}$$

Therefore,

$$J_{\mathrm{RL}}(\theta) = -\beta D_{\mathrm{KL}}(\pi_\theta \| \pi^*_{\mathrm{RL}}) + \beta \log Z,$$

and since $\log Z$ does not depend on θ,

$$\arg\max_\theta J_{\mathrm{RL}}(\theta) \equiv \arg\min_\theta D_{\mathrm{KL}}(\pi_\theta \| \pi^*_{\mathrm{RL}}).$$

□

A.6.2 Action level

Claim. For any state s, maximizing the per-state objective

$$\mathcal{L}_s(\pi) := \mathbb{E}_{a\sim\pi(\cdot|s)}\Big[f(s,a) - \beta \log \frac{\pi(a|s)}{\pi_{\text{ref}}(a|s)}\Big]$$

is equivalent to minimizing the policy-to-target KL

$$D_{\text{KL}}(\pi(\cdot|s)\|\pi_f^*(\cdot|s)),$$

where $\pi_f^*(a|s) \propto \pi_{\text{ref}}(a|s)\exp(f(s,a)/\beta)$; moreover, replacing f by $f - b(s)$ leaves π_f^* unchanged, while using $f = r(s,a)$ generally does not recover the $f = Q(s,a)$ optimum unless $Q(s,a) = r(s,a) + c(s)$.

Proof. Fix a state s and consider the per-state regularized objective

$$\mathcal{L}_s(\pi) := \mathbb{E}_{a\sim\pi(\cdot|s)}\Big[f(s,a) - \beta \log \frac{\pi(a|s)}{\pi_{\text{ref}}(a|s)}\Big],$$

where f is any score function. Define

$$\pi_f^*(a|s) := \frac{1}{Z_f(s)}\,\pi_{\text{ref}}(a|s)\exp\Big(\frac{f(s,a)}{\beta}\Big),$$

$$Z_f(s) = \sum_a \pi_{\text{ref}}(a|s)e^{f(s,a)/\beta}.$$

Then

$$\log \pi_f^*(a|s) = \log \pi_{\text{ref}}(a|s) + \frac{f(s,a)}{\beta} - \log Z_f(s),$$

and hence

$$D_{\text{KL}}\big(\pi(\cdot|s)\,\|\,\pi_f^*(\cdot|s)\big)$$

$$= \mathbb{E}_{a\sim\pi}\big[\log \pi(a|s) - \log \pi_f^*(a|s)\big]$$

$$= \mathbb{E}_{a\sim\pi}\left[\log \frac{\pi(a|s)}{\pi_{\text{ref}}(a|s)} - \frac{f(s,a)}{\beta} + \log Z_f(s)\right].$$

Multiplying by $-\beta$ gives

$$-\beta D_{\text{KL}}(\pi\|\pi_f^*) = \mathcal{L}_s(\pi) - \beta \log Z_f(s).$$

Since $\log Z_f(s)$ is independent of π, maximizing $\mathcal{L}_s(\pi)$ is equivalent to minimizing this policy-to-target KL, and the unique maximizer is π_f^*.

Baseline invariance. Let $f'(s,a) = f(s,a) - b(s)$ for any state-only $b(s)$. Then

$$\pi_{f'}^*(a|s) \propto \pi_{\text{ref}}(a|s)\exp\Big(\frac{f(s,a) - b(s)}{\beta}\Big)$$

$$= \exp\Big(-\frac{b(s)}{\beta}\Big)\pi_{\text{ref}}(a|s)e^{f(s,a)/\beta} \propto \pi_f^*(a|s),$$

so $\pi_{f'}^* = \pi_f^*$. Taking $f = Q$ and $b = V$ yields $\pi^*(a|s) \propto \pi_{\text{ref}}(a|s)e^{A(s,a)/\beta}$, identical to using Q.

Using $r(s,a)$. Replacing f by the immediate reward gives $\pi_r^*(a|s) \propto \pi_{\text{ref}}(a|s)e^{r(s,a)/\beta}$, which coincides with π_Q^* iff $Q(s,a) = r(s,a) + c(s)$ for some state-only $c(s)$. This holds only in special cases (e.g., contextual bandits or $\gamma = 0$); in general MDPs ($\gamma > 0$), Q depends on future returns, so $\pi_r^* \neq \pi_Q^*$. □

B Experimental detail

B.1 Data processing

Training & validation. We filter DAPO-MATH-17K [31] using the following criteria:

1. the final answer is an integer, so that we can reliably extract and compare answers from RL rollouts;
2. the prompt length is at most 1024 tokens;
3. the response length is at most 8192 tokens.

After filtering, we then use the first 10k examples for training and the subsequent 1k examples for validation.

Evaluation. For all benchmarks, we apply no filtering and keep the full dataset.

B.2 Hyper-parameters

Training. All RL algorithms were trained using the same hardware allocation. We sample 8 rollouts per prompt at temperature 1.0, with a maximum response length of 8192 tokens. We use AdamW with weight decay 0.1 and a constant learning rate of 1×10^{-6} with no warmup, and perform one update per batch of 32 prompts. For the main benchmark comparisons, we retain each baseline's KL setting when adding reward budgeting. GRPO uses a reference-policy KL coefficient of 0.001 in both cases.

Validation. We validate every 5 training steps by computing pass@1 accuracy on the validation set. For generation, we use greedy decoding with a maximum response length of 8192 tokens to avoid stochasticity.

Evaluation. We evaluate each trained checkpoint on the selected reasoning benchmarks using 8 responses per prompt at temperature 0.7, with a maximum response length of 8192 tokens. As in Table 2, we report mean accuracy across the eight generations for Qwen3-8B and pass@8 for Llama3.2-3B-Instruct. These are different metrics and should not be compared directly across model families. The table reports point estimates rather than mean $\pm$ std across independently trained models. For the integer-answer benchmarks AMC23, AIME24, and AIME25, we use exact match. For MATH500 and OlympiadBench, whose answers may be symbolic expressions, we use an LLM judge (Qwen3-235B-Instruct) to determine whether a predicted answer matches the ground truth. The judge is instructed to compare *mathematical meaning* rather than surface form: it should ignore differences in LaTeX, whitespace, and notation, and accept mathematically equivalent expressions. It must also enforce any required domain, units, and exactness constraints, while allowing small rounding errors when appropriate. The judge outputs *only* one of two labels: $\boxed{\text{correct}}$ or $\boxed{\text{incorrect}}$.

System message (judge instruction).

```
Judge if the Predicted Answer matches the Ground Truth for the Given Problem.

Be strict on meaning, not format: ignore LaTeX/whitespace/notation; accept mathematically
↪  equivalent forms.
Respect required domain/units/exactness; allow small rounding error.

Reply ONLY with \boxed{correct} or \boxed{incorrect}.
```

User message (template).

```
Given Problem: <question>
Ground Truth: <ground_truth>
Predicted Answer: <pred_answer>
```

B.3 Convergence step detection

In Figure 3, we retrospectively define an operational "convergence step" for each configuration based on a smoothed trajectory rather than raw, potentially noisy evaluations. Concretely, given the sequence of validation accuracies measured every 5 steps, we fit a uni-variate polynomial ($p(x)$) (degree $d = 5$) to (x, y) with x the training step and y the scaled validation accuracy, and evaluate the fitted curve on a dense grid $\{x_g\}_{g=1}^{N_{\text{grid}}}$ ($N_{\text{grid}} = 400$) to obtain $y_g = p(x_g)$. We then identify the earliest grid point x_g that satisfies a sustained plateau condition: (i) local flatness, meaning the absolute change over a forward window of length $w = 0.10 N_{\text{grid}}$ (a fixed fraction of the grid) is small, $|y_{g+w} - y_g| \le \varepsilon_{\text{local}}$; and (ii) global closeness, meaning the remaining gain to the future best value on the fitted curve is small, $\max_{j \ge g}(y_j - y_g) \le \varepsilon_{\text{remain}}$. The tolerances are set relative to the fitted curve's dynamic range $\Delta = \max_g y_g - \min_g y_g$: $\varepsilon_{\text{local}} = \max(\varepsilon_{\text{abs}}, \varepsilon_{\text{rel}}\Delta)$ with $\varepsilon_{\text{abs}} = 0.005$, $\varepsilon_{\text{rel}} = 0.02$, and $\varepsilon_{\text{remain}} = \max(\varepsilon_{\text{abs}}, \rho\Delta)$ with $\rho = 0.10$. To avoid spurious detections, we additionally require that both criteria hold continuously for at least s grid points with $s = 0.06 N_{\text{grid}}$ and a minimum of 10 points. The convergence step is reported as the earliest x_g that meets this sustained condition; we visualize it as a vertical dashed line at $x = x_g$, and optionally plot a plateau guide as the median of the last 10% of fitted values. This procedure measures plateau timing retrospectively on the plotted trajectories. Training uses the validation-plateau stopping rule described above; the numerical tolerances here specify only this retrospective diagnostic.

C Conditional association analysis

Under the set-up in §3.2.1, we further test whether entropy's predictiveness survives after controlling for κ. Specifically, for every run with the principled sharpness factor κ and update step t, we record the entropy $H_{\kappa,t}$ and the validation accuracy $P_{\kappa,t}$. We then set a prediction horizon $\Delta = 10$ steps and regress twice to test whether $H_{\kappa,t}$ predicts $P_{\kappa,t+\Delta}$.

First, we fit a naive regression $P_{\kappa,t+\Delta} = a + \beta_H H_{\kappa,t} + \varepsilon_{\kappa,t}$. Next, we include the configuration-level quantity κ via $P_{\kappa,t+\Delta} = a + \beta'_H H_{\kappa,t} + \beta_\kappa \kappa + \varepsilon_{\kappa,t}$. As shown in Figure 6, the points cluster much more clearly by κ than by H. Consistently, once κ is included as a predictor, it explains most of the predictive power, while the entropy coefficient shrinks substantially ($|\beta'_H| \ll |\beta_H|$). Thus, in this controlled comparison, much of the apparent predictive signal of entropy is accounted for by variation in κ. This comparison describes conditional association in the pooled trajectories. It does not control for training-step effects or serial dependence, and the configurations vary reward scale and KL strength jointly; it therefore does not identify a causal effect of κ or establish entropy as a downstream mediator.

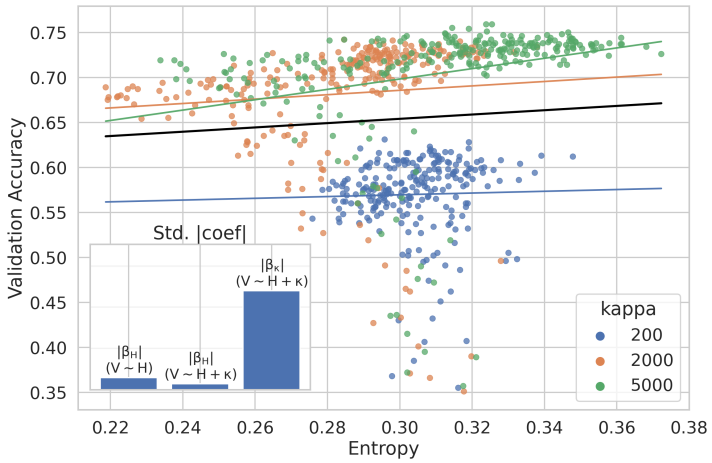


Figure 6: Accuracy evaluated at predictive horizon 10 is plotted against the entropy for three settings $\kappa = 200, 2000, 5000$. Different marker groups indicate different κ. The inset describes the absolute regression coefficients $|\beta_H|$ from OLS linear fits that relate the target metric "Validation Accuracy" V to (i) "Entropy" H alone and (ii) H together with κ, showing the relative importance of H vs. κ.

D Limitations

Our experiments cover verifiable reasoning tasks with predominantly binary rewards; the general-reward analysis assumes additional regularity and does not establish performance for noisy, learned, or open-ended rewards. Each main configuration uses one training seed and $G = 8$, with no group-size sweep. Scores are point estimates; eight evaluation responses measure decoding variability, not training-seed uncertainty. The finite-horizon interventions and regressions do not establish a unique causal mediator of eventual performance. Appendix B.3 retrospectively measures convergence in the Figure 3 intervention trajectories.

E Broader impacts

Improved reasoning performance may benefit mathematical education and scientific problem solving. However, benchmark gains do not establish reliable or calibrated reasoning: incorrect model-generated solutions can mislead users, and the effects of the proposed training changes on other capabilities or safety properties have not been evaluated. Applications therefore require task-specific validation and appropriate human oversight. Training also consumes substantial compute, and this work does not establish a reduction in total energy use.

F Declaration of LLM usage

Appendix B describes the use of Qwen3-235B-Instruct to judge mathematical equivalence and provides the system instruction and user-prompt template. LLM tools also assisted with checking mathematical arguments, refining the proof exposition, and editing the manuscript. The authors take responsibility for the correctness and originality of all mathematical arguments, experimental results, text, and references.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and Section 1 distinguish the finite-G binary-reward result from the conditional general-reward analysis and restrict the empirical claims to the evaluated models and reasoning tasks.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Appendix D discusses binary-reward scope, one training seed per configuration, the fixed rollout-group size, incomplete run-level reporting, and the limits of causal interpretation.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: Appendices A.1, A.3, A.4, A.5, and A.6 state the assumptions and provide the derivations and proofs. Exact objective claims concern the specified idealized score-function estimators rather than the complete clipped optimizer.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [No]

   Justification: Section 5.1 and Appendix B describe the algorithm, data processing, optimizer, validation, and evaluation protocol. Quantitative plateau thresholds, run lengths, checkpoint-selection rules, and complete method-specific clipping configurations are not reported, so exact reproduction of the main table is not fully specified.

   Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification: The experiments use existing datasets and models. Code will be released upon approval at `https://github.com/Qwen-Applications/RewardBudget`; public code and a complete reproduction package are not available with this version.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [No]

   Justification: Section 5.1 and Appendix B specify the data split, rollout and evaluation settings, AdamW with weight decay 0.1, and the constant learning rate 10^{-6}. Complete method-specific clipping configurations and quantitative stopping and checkpoint-selection rules remain unreported.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [No]

   Justification: Each main configuration uses one training seed, and the main table reports point estimates without uncertainty across independent training runs. The eight responses per evaluation prompt concern decoding variability and do not provide training-seed error bars, as discussed in Appendices D and B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [No]

   Justification: Appendix B states that the methods use a common hardware allocation. Hardware specifications, per-run execution times, and the total compute used for the reported and preliminary experiments are not disclosed.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The authors confirm that the research conforms to the NeurIPS Code of Ethics. The study uses existing reasoning benchmarks and involves no human-subject experiments; its scope and potential impacts are discussed in the Limitations subsection and Appendix E.

Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Appendix E discusses potential benefits for mathematical and scientific applications, risks from incorrect model outputs, and the limits of deployment and energy-efficiency claims.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This version does not release new pretrained models or scraped datasets. It studies a training algorithm using existing models and reasoning benchmarks, with code release subject to the required approval.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.

- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [No]

    Justification: The datasets, models, and baseline methods are credited in Section 5.1 and the references. A complete asset-by-asset record of versions, licenses, and terms of use is not explicitly provided in this paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: No new dataset, model weights, or code package is released with this version of the paper. The planned implementation release is subject to the required approval and will require its accompanying documentation.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The research does not involve crowdsourcing or human-subject experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: The research does not involve human participants, so human-subject risk disclosure and IRB approval are not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: Appendix B identifies Qwen3-235B-Instruct as the mathematical-equivalence judge and provides its system instruction and user prompt. Appendix F also discloses LLM assistance with mathematical checking, proof exposition, and manuscript editing.

    Guidelines:

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