RETHINKING PROBABILITY-BASED REINFORCEMENT LEARNING FROM POSTERIOR CONCENTRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Verifier-free reinforcement learning with probability-based rewards offers a promising way to train LLMs on general reasoning tasks where external verifiers are unavailable. Yet the reliability of these rewards, especially in long-horizon reasoning, remains underexplored. This work identifies a length-dependent failure mode of probability rewards, which we call the Posterior Concentration Phenomenon (PCP). We show that the probability of a reference answer conditioned on a reasoning trace often collapses to a low-variance interval as the trace becomes lengthy. This phenomenon results in nearly indistinguishable rewards, which, under GRPO-based settings, makes probability-based policy optimization unstable and inefficient. Motivated by this, we propose *Reinforcement Learning with Concentration-aware Posterior Rewards* (**RLCPR**), a verifier-free RL framework to explicitly account for PCP for better optimization stability and token efficiency. It has two components: **uncertainty-aware data sampling**, which reduces concentration-prone rollouts before generation, and **concentration-aware regularization**, which penalizes unnecessarily long traces when posterior rewards collapse. Extensive experiments show that, alongside higher token efficiency, RLCPR outperforms the state-of-the-art verifier-free RL baseline by up to 4.0% on six of seven benchmarks, including general-domain and mathematical reasoning challenges. Source code will be released upon acceptance.

1 INTRODUCTION

Reinforcement learning (RL) has become a prominent solution to advance the reasoning capabilities of large language models (LLMs) (Ouyang et al., 2022). By optimizing models on sampled reasoning trajectories, RL lets a model learn from its own successes and failures, and has delivered strong gains in mathematical reasoning (Hu et al., 2026; Zeng et al., 2025; Yu et al., 2026), code generation (Cui et al., 2025), and complex decision-making tasks.

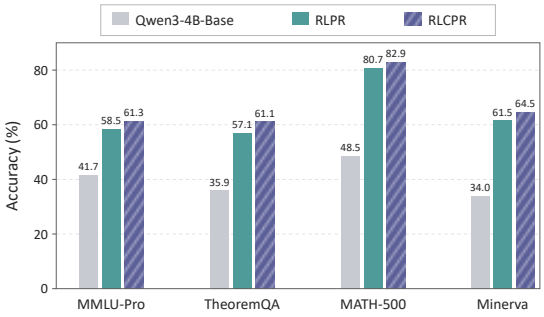


Figure 1: RLCPR performance on representative general-domain and mathematical benchmarks.

However, many existing RL approaches for LLM reasoning rely on verifiable rewards, where correctness can be easily checked by pre-defined rules or by external verifier-purposed models (Lambert et al., 2024; Chen et al., 2026). This requirement limits their applicability to broader reasoning settings where ground-truth verification is unavailable, ambiguous, or expensive. Recently, probability-based rewards have emerged as a promising verifier-free alternative (Yu et al., 2025; Liu et al., 2025; Zhou et al., 2025). Instead of using an external verifier, these methods directly use the LLM's own probability estimates to construct rewards. A representative formulation is the posterior reward, which scores a reasoning trace (z) by the model's posterior probability of generating the ground-truth answer (y^*) after conditioning on z, i.e., $R_\theta(z) = P(y^* \mid x, z; \theta)$. Intuitively, this posterior probability-based reward encourages the model to learn reasoning traces associated with a higher probability of

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

the reference answer. It provides dense, self-contained supervision that extends beyond strictly verifiable domains and requires no external model during training.

Despite their practical appeal, the mechanism behind posterior probability-based rewards is underexplored. It is unclear whether these rewards can reliably distinguish high- versus low-quality reasoning traces. This question is particularly important for group-relative RL algorithms such as GRPO (Shao et al., 2024), where training depends on the difference in the rewards across multiple rollouts for the same prompt. Recent evidence suggests a reason for concern. Welch et al. (2026) show that, once a model has produced a reasoning trace, the likelihood it assigns to an answer increasingly reflects consistency with that trace rather than the model's belief in the answer itself. If posterior rewards inherit this behavior, a sufficiently long trace may lock in its answer regardless of whether the reasoning is sound.

Building on this concern, this work identifies a length-dependent failure mode of posterior rewards, which we call the *Posterior Concentration Phenomenon* (PCP). Our exploration experiments show that, as reasoning traces grow longer, posterior rewards for traces ending in correct and incorrect answers become less distinguishable, while their cross-rollout variance also contracts. We further connect this empirical behavior to recent results on log-likelihood concentration in long LLM generations (Bell et al., 2024; Yang et al., 2026). Specifically, as sequence likelihoods concentrate, it causes the posterior probability of the reference answer to depend less on the specific reasoning trajectory. Under a mild condition motivated by empirical evidence on LLM branching factors (Yang et al., 2026), we show that different reasoning traces become increasingly likely to receive similar posterior rewards as the considered range of reasoning lengths expands. This reveals a fundamental limitation of posterior rewards for group-relative optimization: as their ability to reflect reasoning quality diminishes, the reward differences that GRPO relies on no longer indicate which rollouts are better, leaving the policy without an informative optimization direction. Worse still, since rollouts tend to grow longer as training proceeds, the training dynamics are pushed deeper into the concentrated regime over time.

PCP reframes how posterior-based RL should be designed. If posterior rewards lose discriminative power as traces lengthen, a training framework should (i) avoid spending generation budget on prompts likely to produce concentrated rollout groups, and (ii) prevent the policy from drifting into the long-trace regime where posterior rewards no longer reflect reasoning quality. Existing methods address neither directly. RLPR (Yu et al., 2025), for instance, discards rollout groups with low reward variance, but it does so without characterizing when or why such groups arise. Moreover, because filtering occurs only after rollouts are generated and scored, the generation cost of the discarded groups is wasted rather than avoided. Motivated by our analysis, we propose **R**einforcement **L**earning with **C**oncentration-aware **P**osterior **R**ewards (**RLCPR**), a verifier-free RL framework that explicitly accounts for PCP for better optimization stability and token efficiency. RLCPR contains two major components: (1) an uncertainty-aware data sampling method to efficiently sample informative training data with a lower risk of PCP, (2) a regularization mechanism that explicitly accounts for PCP and penalizes unnecessarily lengthy reasoning traces. Rather than replacing the posterior rewards, RLCPR improves their reliability by preserving useful posterior information while reducing the adverse effect of long-horizon concentration. Overall, RLCPR outperforms RLPR by up to 4.0 percentage points on six of seven benchmarks, including MMLU-Pro, TheoremQA, WebInstruct, MATH-500, Minerva, and AIME24. Our contributions are summarized as follows:

- We identify the Posterior Concentration Phenomenon (PCP), a length-dependent loss of answer-probability discriminability in long LLM reasoning traces, and connect it to the reward collapse in probability-based RL.
- We provide the first theoretical formulation of posterior-reward collapse and support it empirically through diminishing reward separation and contracting reward variance.
- We propose RLCPR, which explicitly accounts for PCP through uncertainty-aware data sampling and concentration-aware regularization.

2 Related Work

Reasoning and confidence reliability. Prior work examines whether language models' confidence reflects answer correctness (Jiang et al., 2021; Kadavath et al., 2022). Explicit reasoning complicates

this relationship. While Yoon et al. (2026) find that verbalized confidence can become better calibrated as chain of thoughts (CoT) unfolds, Welch et al. (2026) show that answer-token probabilities in vision-language models can increasingly reflect consistency with the generated reasoning rather than correctness. This concern is reinforced by studies of reasoning faithfulness: CoT explanations can rationalize hidden biases (Turpin et al., 2023), and final predictions do not always depend on the stated reasoning (Lanham et al., 2023). Moreover, prior beliefs can induce confirmation bias in both rationale generation and answer prediction (Wan et al., 2025). These findings motivate examining whether confidence reliably distinguishes reasoning quality. Our work studies this question in posterior-based RL, where rewards must remain *relatively discriminative* across reasoning traces, and identifies a length-dependent loss of this property through PCP.

Verifier-free reinforcement learning. Probability-based RL uses the model's own confidence in reference answers to provide supervision without external verifiers. JEPO (Tang et al., 2025) and NOVER (Liu et al., 2025) construct training signals from answer log-probabilities and token-level perplexities, respectively. More directly related to our work, VeriFree (Zhou et al., 2025) rewards reasoning traces using posterior probabilities of reference answers, while RLPR (Yu et al., 2025) extends this approach to non-verifiable domains through length normalization and reward debiasing. These methods demonstrate the promise of probability-based supervision, but its reliability for long reasoning traces remains underexplored. We address this gap by analyzing posterior concentration and introducing RLCPR, which mitigates its effects through uncertainty-aware data sampling and concentration-aware regularization.

3 PRELIMINARIES

3.1 POSTERIOR PROBABILITY-BASED RL

Probability-based RL methods construct rewards from the model's conditional probabilities of the reference-answer tokens. It can be generally formulated as:

$$R_\theta^{\text{Post}}(z) = \tilde{P}(Y = y^* \mid Z = z; x; \theta), \tag{1}$$

where z is the reasoning sequence generated by the LLM, y^* is the ground-truth answer, x is the input prompt, and Z and Y are the random variables of reasoning sequence and prediction sequence, respectively. Intuitively, Eq. (1) represents the model's posterior belief in the reference answer conditioned on a reasoning trace. Existing methods instantiate this posterior-derived signal through different token-level aggregations, *i.e.* $r = f(R_\theta(z))$ with different f, such as sequence likelihood in VeriFree (Zhou et al., 2025) and token-level probability mean in RLPR (Yu et al., 2025).

Rewards such as Eq. (1) are used in RL training with methods such as GRPO (Shao et al., 2024). For each input q, the old policy model samples N candidate outputs $\{o_i\}_{i=1}^N$ to form a group. Their corresponding rewards $\{r_i\}_{i=1}^N$ are then normalized within the group, *i.e.* $A_i = \frac{r_i - \mu}{\sigma}$, to compute relative advantages. It measures each candidate's quality with respect to its siblings. The GRPO objective is

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\{o_i\}\sim\pi_{\theta_{\text{old}}}(q)}\left[\frac{1}{N}\sum_{i=1}^{N}\Big\{\min\big[s_1 A_i,\ s_2 A_i\big] - \beta\,\mathbb{D}_{\text{KL}}\big[\pi_\theta \,\|\, \pi_{\text{ref}}\big]\Big\}\right], \tag{2}$$

where $s_1 = \frac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}$ and $s_2 = \text{clip}\Big(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)},\ 1-\epsilon,\ 1+\epsilon\Big)$ with the clipping function that limits the interval to $[1-\epsilon, 1+\epsilon]$.

To improve optimization efficacy, RLPR (Yu et al., 2025) filters low-variance groups. Let $\sigma(g)$ denote the standard deviation of group g's rewards; the group is retained only if

$$\sigma(g) > \text{EMA}_{j\in\mathcal{G}}[\sigma(j)] \times \beta, \tag{3}$$

where $\mathcal{G}$ denotes all the sampled groups and $\text{EMA}_{j\in\mathcal{G}}$ is the exponential moving average of all the within-group standard deviations. As filtering is performed after the rollouts are generated and scored, low-information groups still incur the full generation cost. In Sec. 4, we show that such low-variance posterior rewards can arise systematically from posterior concentration in long reasoning traces.

3.2 Entropy and Log-Likelihood Concentration for LLMs

Let $\widetilde{P}(o_t \mid [x, o_{1:t-1}]; \theta)$ denote the normalized autoregressive distribution induced by vocabulary truncation during decoding (Holtzman et al., 2019; Hewitt et al., 2022). Bell et al. (2024) establish an un-equipartition property showing that the length-normalized NLL of a model-generated sequence converges in probability to its length-normalized realized entropy:

$$\lim_{N\to\infty} \Pr\left(\left|-\frac{1}{N}\log \widetilde{P}(o_{1:N} \mid x;\theta) - \frac{1}{N}h_{\text{realized}}(o_{1:N})\right| > \epsilon\right) = 0, \tag{4}$$

where $h_{\text{realized}}(o_{1:N}) := \sum_{t=1}^{N} \widetilde{H}\big(O_t \mid [x, o_{1:t-1}]; \theta\big)$ is the *realized entropy* aggregating the next-token entropy along the trajectory. Its expectation is the deterministic full-sequence entropy:

$$\mathbb{E}_{o_{1:N}}\left[h_{\text{realized}}(o_{1:N})\right] = \widetilde{H}(O_{1:N} \mid x;\theta) = \sum_{t=1}^{N} h_{x,\theta}(t), \tag{5}$$

where $h_{x,\theta}(t) = \mathbb{E}_{o_{1:t-1}}[\widetilde{H}(O_t \mid [x, o_{1:t-1}]; \theta)]$.

Building on this connection, Yang et al. (2026) empirically show that long-sequence likelihoods can be characterized by the deterministic sequence entropy, motivating the approximation

$$-\frac{1}{N}\log \widetilde{P}(o_{1:N} \mid x;\theta) \approx \frac{1}{N}\widetilde{H}(O_{1:N} \mid x;\theta). \tag{6}$$

Consequently, sufficiently long sequences of the same length tend to have similar NLLs despite following different generation trajectories. We refer to this tendency as *log-likelihood concentration*.

Yang et al. (2026) further characterize the effective generative breadth through the *branching factor*:

$$B_N(x;\theta) := \exp\left(\frac{1}{N}\widetilde{H}(O_{1:N} \mid x;\theta)\right). \tag{7}$$

A larger value indicates a more diffuse generative distribution. Its empirically observed decrease with generation length suggests that later positions become increasingly predictable and motivates the entropy condition used in our PCP analysis.

4 Posterior Concentration Phenomenon

This section rethinks the concentration phenomenon of LLMs from the perspective of posterior probabilities. We are guided by two questions: *Given two reasoning traces generated by the same model for the same prompt, does the posterior probability reflect their reasoning quality? If not, under what conditions does it become less informative for GRPO optimization?* We explore with an empirical study in Sec. 4.1, before introducing our proposed Posterior Concentration Phenomenon (PCP) in Sec. 4.2.

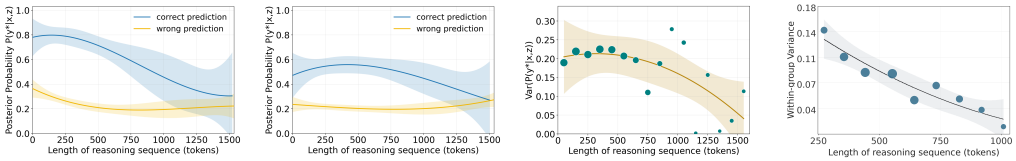
4.1 Warm-up: Exploration Experiments

We begin by examining posterior rewards over progressively longer reasoning traces. Fig. 2 shows two consistent signatures on the benchmarks MMLU-Pro (Wang et al., 2024) and GPQA-Diamond (Rein et al., 2023). First, posterior rewards for traces ending in correct versus incorrect answers become less distinguishable at longer lengths (Figs. 2a and 2b, blue vs. yellow). Second, posterior-reward dispersion contracts with length: the overall reward variance decreases (Fig. 2c), and the variance among rollouts within the same group also steadily declines (Fig. 2d). Together, these results show that long reasoning traces provide less discriminative rewards within the groups used for relative policy optimization. We refer to this length-dependent loss of discriminative power as the **Posterior Concentration Phenomenon (PCP)**. See Appx. E.1 for the details of this exploration experiment.

4.2 Posterior Concentration Phenomenon

Let Z and Y denote the random variables representing the reasoning and answer sequences, where the LLM first generates the reasoning trace and then the answer. For a reasoning trace z and reference answer y^*, the posterior reward can be separated into two terms:

$$\log \widetilde{P}(y^* \mid x, z;\theta) = \log \widetilde{P}(z, y^* \mid x;\theta) - \log \widetilde{P}(z \mid x;\theta), \tag{8}$$



(a) MMLU-Pro (b) GPQA-Diamond (c) MMLU-Pro (Var.) (d) GPQA-Diamond (Within-group Var.)

Figure 2: **Posterior rewards become less discriminative as reasoning length grows.** On MMLU-Pro and GPQA-Diamond, posterior rewards for traces ending in correct and incorrect answers approach one another at longer lengths (a, b). Also, the reward variance (c) and within-group variance (d) contract in the long-length regime. The shaded area represents the 95% confidence interval. These experimental results point to an effect we named as **Posterior Concentration Phenomenon**, where posterior probability collapses to a similar value as the reasoning length grows.

where $\widetilde{P}$ represents the normalized generative distribution. Let $n = |z|$ and $m = |y^*|$ denote the lengths of the reasoning and answer sequences respectively. We define the residual of log-likelihood concentration as $\epsilon_n(o_{1:n}) := -\log \widetilde{P}(o_{1:n} \mid x;\theta) - \widetilde{H}_n(x;\theta)$, based on the approximation in Eq. (6), where $\widetilde{H}_n(x;\theta) = \widetilde{H}(O_{1:n} \mid x;\theta)$ denotes the expected entropy over a length-n sequence. By substituting the residuals into Eq. (8), we obtain

$$\log \widetilde{P}(y^* \mid x, z;\theta) = -\left[\widetilde{H}_{n+m}(x;\theta) - \widetilde{H}_n(x;\theta)\right] + \left[\epsilon_n(z) - \epsilon_{n+m}(z, y^*)\right]. \tag{9}$$

Eq. (9) implies that a log-posterior probability, at a sequence-level, can be decomposed into two deterministic entropies and the corresponding residuals pushing them away from these approximations. Now, consider two independent traces z^1, z^2 with $|z^1| = n + s, |z^2| = n$, where n is sufficiently large and s is a small positive integer. The log ratio between two posteriors can then be derived from Eqs. (8) and (9) as:

$$\log \frac{\widetilde{P}(y^* \mid x, z^1;\theta)}{\widetilde{P}(y^* \mid x, z^2;\theta)} = -\left[\widetilde{H}_{n+s+m}(x;\theta) - \widetilde{H}_{n+s}(x;\theta)\right] + \left[\widetilde{H}_{n+m}(x;\theta) - \widetilde{H}_n(x;\theta)\right] + \rho_{n+s}(z^1) - \rho_n(z^2), \tag{10}$$

where $\rho_n(z) = \epsilon_n(z) - \epsilon_{n+m}(z, y^*)$ is the length-induced residual shift that measures the differences between log-likelihood concentrations before and after generating the answer y^*. Recall from Eq. (5) that $\widetilde{H}_n(x;\theta) = \sum_{t=1}^{n} h_{x,\theta}(t)$. If we define the answer-window entropy after a length-n reasoning trace as $W_n = \widetilde{H}_{n+m}(x;\theta) - \widetilde{H}_n(x;\theta) = \sum_{t=n+1}^{n+m} h_{x,\theta}(t)$, then Eq. (10) simplifies into:

$$\log \frac{\widetilde{P}(y^* \mid x, z^1;\theta)}{\widetilde{P}(y^* \mid x, z^2;\theta)} = \underbrace{W_n - W_{n+s}}_{\text{length-induced entropy shift}} + \underbrace{\rho_{n+s}(z^1) - \rho_n(z^2)}_{\text{concentration residuals}}. \tag{11}$$

Eq. (11) decomposes the unequal-length posterior difference into two parts. The first term, $W_n - W_{n+s}$, is a deterministic, length-induced entropy shift. The second term, $\rho_{n+s}(z^1) - \rho_n(z^2)$, captures the trace-specific residuals of the log-likelihood approximation. For equal-length traces, *i.e.* $s = 0$, the deterministic entropy shift cancels exactly, and the log-posterior difference depends entirely on the trace-specific residual difference. For unequal-length traces, we additionally characterize the entropy shift $W_n - W_{n+s}$ by the following lemma.

Define the length-averaged token entropy as $\bar{h}_{x,\theta}(n) := \widetilde{H}_n(x;\theta)/n = \frac{1}{n}\sum_{t=1}^{n} h_{x,\theta}(t)$. Notably, $\exp(\bar{h}_{x,\theta}(n))$ is exactly the branching factor defined in Eq. (7). Prior work empirically observes that this quantity generally decreases as generation progresses (Yang et al., 2026). Motivated by this, we consider eventual non-increase of $\bar{h}_{x,\theta}(n)$ as a sufficient condition and establish the following lemma, with its proof deferred to Appx. A.

Lemma 4.1 (Asymptotic Answer-Window Stability) *If $\bar{h}_{x,\theta}(n)$ is eventually non-increasing, then for any fixed positive integers m and s, and any $\epsilon > 0$,*

$$\lim_{N\to\infty} \frac{1}{N}\left|\left\{n \in \{1, \ldots, N\} : |W_{n+s} - W_n| > \epsilon\right\}\right| = 0. \tag{12}$$

Intuitively, Lem. 4.1 states that, as the length range grows, large differences in answer-window entropy with a length gap s becomes more rare. Notably, this does not require the token-level entropy to vary smoothly. Local fluctuations with particularly high-entropy tokens may persist, but non-negligible shifts on answer-window entropies become asymptotically sparse. Combining Lem. 4.1 with Eq. (11), we obtain the following PCP characterization:

Proposition 4.1 (Posterior Concentration Phenomenon.) *Let Z_n denote the random reasoning sequence generated by the model conditioned on having length n. Let $L_N \sim \mathrm{Unif}(1, \ldots, N)$ be an auxiliary random reasoning length. Suppose that $\bar{h}_{x,\theta}(n)$ is eventually non-increasing. Given a fixed $s > 0$, if, for any $\delta > 0$, $\mathbb{P}\left(|\rho_{n+s}(Z_{n+s}) - \rho_n(Z_n)| > \delta\right) \to 0$ as $n \to \infty$, then, for any $\epsilon > 0$,*

$$\lim_{N\to\infty} \mathbb{P}\left(\left|\log \frac{\widetilde{P}(y^* \mid x, Z_{L_N+s}; \theta)}{\widetilde{P}(y^* \mid x, Z_{L_N}; \theta)}\right| > \epsilon\right) = 0, \tag{13}$$

where the probability is taken jointly over the auxiliary length L_N and the random reasoning sequences Z_{L_N+s} and Z_{L_N}. See proof in Appx. B.

Above, L_N serves as a probabilistic counterpart to the deterministic length index n in Lem. 4.1: uniformly sampling L_N converts Lem. 4.1's zero-density statement over reasoning lengths into an equivalent probability statement. Intuitively, Prop. 4.1 suggests that, as the considered range of reasoning length grows, posterior probabilities for independently sampled traces of similar lengths become more likely to take similar values when their residual differences are close. Note that the sufficient condition of residual difference is motivated by the empirical observations from Yang et al. (2026). See Appx. G for further discussion. Importantly, this does not imply lower model confidence; rather, the posterior becomes less discriminative about the quality of the underlying reasoning trace. This observation complements recent findings that reasoning can distort probability-based confidence signals (Welch et al., 2026; Fu et al., 2025; Xiong et al., 2026). It uncovers the limitation of answer posterior probabilities as supervision signals: internal confidence or consistency does not necessarily provide reliable discrimination of reasoning quality.

From an RL perspective, this loss of discriminability weakens the learning signal provided by posterior rewards: long rollouts may receive similar rewards despite differences in their reasoning quality, potentially making the relative advantages less informative about reasoning quality while still incurring substantial generation cost. Since PCP is a probabilistic effect, this motivates a PCP-aware framework with two complementary directions: (1) preferentially sample training examples that are less prone to posterior concentration, and (2) discourage unnecessarily long reasoning once posterior difference collapses. We introduce these two components in Sec. 5.

5 RLCPR

To account for posterior concentration during policy optimization, we propose RLCPR, a novel verifier-free post-training framework with two core components: uncertainty-aware data sampling (UDS) and concentration-aware regularization (CR).

Uncertainty-aware data sampling (UDS). Discarding less informative groups with low standard deviation, as done by Yu et al. (2025), is inefficient. We propose to use uncertainty-aware data sampling to reduce unnecessary rollouts that may potentially suffer from PCP. Our sampling strategy is motivated by an empirical association: questions whose rollout groups exhibit posterior concentration tend to have higher direct-answer uncertainty during training. This suggests using uncertainty as a proxy for concentration risk and preferentially retaining lower-uncertainty examples. Appx. C presents the diagnostic experiment about this relation between uncertainty and concentration.

Specifically, for each question x, we concatenate a suffix q, and define the model's uncertainty as the average token entropy of the k tokens after the sequence $[x, q]$. In this paper, our suffix q is *"In short, the answer is "*. This uncertainty can be formulated as:

$$U_\theta(x) = \frac{1}{k} \sum_{t=1}^{k} \tilde{H}\big(O_t \mid [x, q, \hat{o}_{1:t-1}]; \theta\big), \tag{14}$$

where $\hat{o}_{1:t-1}$ is the output sequence until position $t-1$ under greeding decoding strategy, *i.e.*, $\hat{o}_t = \arg\max_{v \in V} \tilde{P}\left(v \mid [x, q, \hat{o}_{1:t-1}]; \theta\right)$ with V as the vocabulary. We refer to this quantity as

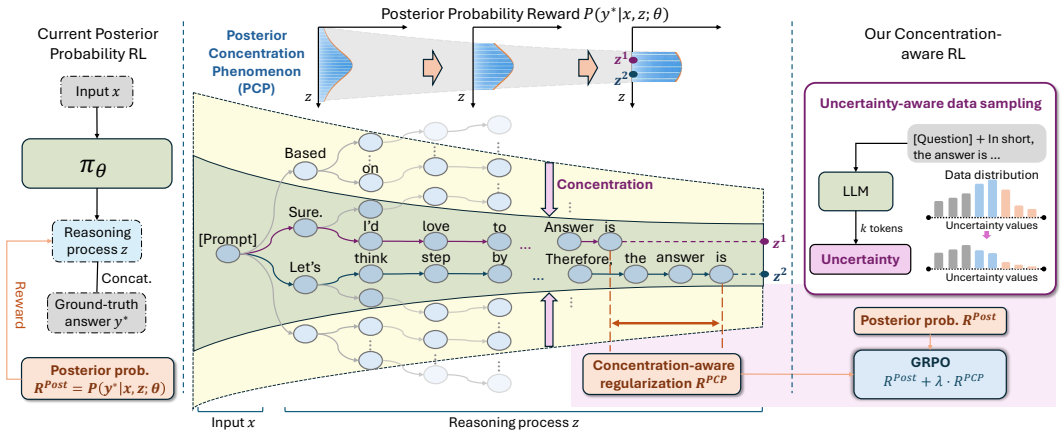


Figure 3: **Overview of RLCPR.** RLCPR is a PCP-aware training framework with two components: Uncertainty-aware data sampling (UDS) reshapes the data distribution according to direct-answer uncertainty, reducing rollouts that are more prone to posterior concentration; Concentration-aware regularization (CR) detects concentrated rollout groups and introduces a length-aware penalty to discourage unnecessarily long reasoning traces.

direct-answer uncertainty. During the experiments, we use a buffer storage to save the uncertainty values and update this buffer every f GRPO steps. This process is equivalent to using fixed model parameters $\hat{\theta}$ to estimate the uncertainty $U_\theta(x)$ with $U_{\hat{\theta}}(x)$ and periodically updating $\hat{\theta}$ after every f steps. It achieves high computational efficiency while preserving reliable approximation to the model's uncertainty to different question-response pairs.

Based on the proposed uncertainty measure, we introduce two quantile-based thresholds τ_1 and τ_2 with $\tau_0 = 0 \leq \tau_1 \leq \tau_2 \leq 1 = \tau_3$ to divide the question batch into three bins b_1, b_2, b_3. More precisely, this process can be represented with the following expression:

$$\mathbb{1}\Big(x \in b_i\Big) \equiv \mathbb{1}\Big(\tau_{i-1} \leq P_{x' \sim \mathcal{D}}\big(U_\theta(x) > U_\theta(x')\big) < \tau_i\Big), \tag{15}$$

where $\mathbb{1}(\cdot)$ is the binary indicator function. Then, we use the filtering rates $\omega_1, \omega_2, \omega_3$ to randomly remove the data in b_1, b_2, b_3. This operation is equivalent to modifying the sampling strategy such that the data sampling distribution is related to the model uncertainty level. Unlike post-generation reward filtering, UDS filters questions before generating full reasoning traces. Its uncertainty estimates require only short direct-answer completions and are cached across training steps, amortizing the estimation cost. Appx. D provides the complexity analysis.

Concentration-aware regularization (CR). We also propose a concentration-aware regularizer for GRPO-based training. When concentration happens, the posterior-based rewards no longer reflect the quality of reasoning processes, so we argue that the model should learn to reduce the reasoning lengths outright. Inspired by Team et al. (2025), we adopt a length penalty to penalize relatively longer reasoning sequences and encourage relatively shorter reasoning sequences within a group. Different from Team et al. (2025), our length penalty is used only when concentration happens. Specifically, consider a gap of posterior probabilities for a group G as follows:

$$\Delta G = \max_{z^1, z^2 \in G} \Big| \tilde{P}(y_G^* \,|\, x_G, z^1; \theta) - \tilde{P}(y_G^* \,|\, x_G, z^2; \theta) \Big|, \tag{16}$$

where x_G, y_G^* are the input question and reference answer corresponding to group G. We then propose the following penalty:

$$R_\theta^{\text{PCP}}(z) = -\mathbb{1}\Big(l_{\min} > \delta \text{ and } \Delta G_z < \alpha\Big) \cdot \frac{|z| - l_{\min}}{\max\{l_{\max} - l_{\min}, 1\}}, \tag{17}$$

where G_z denotes the group corresponding to z in GRPO, $l_{\min} = \min_{z' \in G_z} |z'|$, and $l_{\max} = \max_{z' \in G_z} |z'|$. Intuitively, Eq. (17) defines a binary indicator to detect the groups with PCP and imposes the length-aware penalty on these groups to encourage shorter responses. Concentration occurs if all the reasoning sequences are sufficiently long (with respect to δ) and the posterior probabilities are similar

Table 1: **Overall performance on seven reasoning benchmarks.** Avg@k denotes accuracy estimated using k sampled responses per question. Compared with same-backbone RLPR, RLCPR outperforms RLPR on six of seven benchmarks and improves average accuracy by 1.8% while achieving higher training token efficiency. WebInst. denotes the held-out evaluation subset of WebInstruct.

Method	Backbone	Verifier	MMLU-Pro Avg@2	GPQA-Dia. Avg@4	TheoremQA Avg@2	WebInst. Avg@2	MATH-500 Avg@2	Minerva Avg@2	AIME24 Avg@16	All
Base	Qwen2.5-7B	-	45.3	32.4	41.4	60.4	63.0	37.6	6.5	40.9
RLVR		Rule	55.1	36.2	52.2	75.3	76.5	54.9	**17.7**	52.6
Gen. Reasoner		Model	55.4	37.4	52.1	74.5	77.0	51.7	16.0	52.0
VeriFree		Free	53.8	36.7	47.6	72.5	73.5	49.0	12.5	49.4
RLPR		Free	56.0	37.6	55.4	75.5	78.0	56.5	16.3	53.6
Base	Qwen3-4B-Base	-	41.7	25.1	35.9	54.7	48.5	34.0	5.0	35.0
RLPR		Free	58.5	**43.4**	57.1	78.0	80.7	61.5	14.0	56.2
RLCPR		Free	**61.3** (+2.8)	40.7	**61.1** (+4.0)	**78.9** (+0.9)	**82.9** (+2.2)	**64.5** (+3.0)	16.3 (+2.3)	**58.0** (+1.8)

(with respect to α). The two thresholds α and δ are decided at runtime. Specifically, we compute quantiles over the training batch to adapt to the variation of policy models and data distributions:

$$\delta = Q_{p_\delta}(\{|z| : z \in G,\ G \in \mathcal{B}\}), \qquad \alpha = Q_{p_\alpha}(\{\Delta G : G \in \mathcal{B}\}), \tag{18}$$

where $\mathcal{B}$ denotes the rollout groups in the current training batch and $Q_p(\cdot)$ denotes the empirical p-quantile. The final concentration-aware (CA) reward for a given rollout is then defined as the weighted sum of the posterior-based reward and the PCP-aware regularization, *i.e.*

$$R_\theta^{\text{CA}}(z) = R_\theta^{\text{Post}}(z) + \lambda R_\theta^{\text{PCP}}(z). \tag{19}$$

Eq. (19) suggests a straightforward learning objective: When little information can be extracted solely from the posterior probability, the model should learn to shorten the length. Building on our empirical experience, we suggest a cold start for the PCP-aware penalty, i.e. a lower λ at the beginning and gradually increasing the weight.

6 Experiments

6.1 Experimental settings

Model and implementation. We evaluate RLCPR on Qwen3-4B-Base (Yang et al., 2025). Our implementation builds on the official RLPR (Yu et al., 2025) codebase in `verl` (Sheng et al., 2025) but we exclude the standard-deviation filtering. For UDS, we set $k = 3$, $f = 20$, and $(\tau_1, \tau_2) = (0.3, 0.7)$ with linear annealing during the first 200 steps; the filtering rates are $(\omega_1, \omega_2, \omega_3) = (0.2, 0.45, 0.7)$, where data with higher uncertainty are associated with lower sampling rates. For CR, p_δ and p_α are set to 0.8 and 0.2, respectively. We train for 600 steps to achieve an output-token budget comparable to RLPR, with $\lambda = 0$ for the first 160 steps and 0.5 thereafter. Other hyperparameters follow RLPR. Implementation details and training prompt can be found in Appx. E.3 and E.4.

Baselines. Our primary baseline is RLPR with the same backbone. We additionally report the Qwen2.5-7B results excerpted from Yu et al. (2025), including RLVR (Hu et al., 2026), General Reasoner (Ma et al., 2025), VeriFree (Zhou et al., 2025), and RLPR (Yu et al., 2025).

Datasets. Following RLPR, we train on 77K non-mathematical reasoning prompts from Ma et al. (2025), filtered by GPT-4.1 (OpenAI, 2025). We evaluate on four general-domain benchmarks– MMLU-Pro (Wang et al., 2024), GPQA-Diamond (Rein et al., 2023), TheoremQA (Chen et al., 2023), and the held-out WebInstruct subset (Ma et al., 2025); and three mathematical benchmarks– MATH-500 (Lightman et al., 2023), Minerva (Lewkowycz et al., 2022), and AIME24 (Zhang & Math-AI, 2024).

6.2 Main results

Table 1 summarizes performance on seven reasoning benchmarks. On Qwen3-4B-Base, RLCPR improves average accuracy from 56.2% to 58.0%, outperforming RLPR on six of seven benchmarks. It gains 2.8, 4.0, and 0.9 percentage points on MMLU-Pro, TheoremQA, and WebInstruct, respectively. Despite training exclusively on non-mathematical prompts, RLCPR also improves MATH-500, Minerva, and AIME24 by 2.2, 3.0, and 2.3 points, demonstrating benefits beyond the training domain.

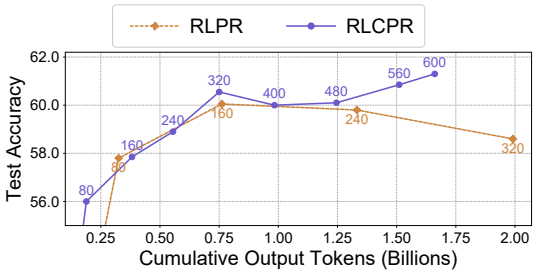


Figure 4: **Rollout-token efficiency.** RLCPR achieves higher accuracy on MMLU-Pro with fewer cumulative output tokens. Numbers on the curves denote training steps.

Table 2: Ablation of UDS and CR.

Method	MATH-500	Minerva	AIME24
Baseline	78.9	57.8	14.4
+ CR	79.5	59.7	15.0
+ UDS	76.9	56.6	14.6
RLCPR	**82.9**	**64.5**	**16.3**

Table 3: Ablation of CR cold start and UDS linear-annealing scheduler.

Method	MMLU-Pro	TheoremQA	WebInstruct
w/o CR cold start	56.4	52.5	71.0
w/o UDS scheduler	60.2	59.1	78.2
RLCPR	**61.3**	**61.1**	**78.9**

RLCPR also improves rollout-token efficiency. As shown in Fig. 4, at 320 steps, RLCPR uses 0.75 billion rollout tokens, 62.3% fewer than RLPR's 1.99 billion, while still achieving higher accuracy; even after 600 steps, it uses only 1.66 billion tokens. See Appx. D for the complexity analysis. RLCPR nevertheless shows a 2.7% decrease on GPQA-Diamond relative to RLPR. This may partly arise because favoring lower-uncertainty examples and shorter traces could limit learning opportunities for challenging problems requiring extended reasoning. Although this explanation warrants further investigation, the benchmark-specific decrease does not alter the overall gains in average accuracy and rollout-token efficiency.

6.3 Ablation Study

UDS and CR effect. Table 2 evaluates the individual and joint effects of UDS and CR. CR alone provides modest gains, whereas UDS alone slightly degrades performance on MATH-500 and Minerva, possibly because it filters out some difficult but informative training instances. Importantly, UDS is not a static difficulty-based filter: it periodically re-estimates uncertainty under the current policy (every $f = 20$ steps) and dynamically reshapes the training distribution. Its effectiveness therefore depends on how the policy evolves. Without CR, the retained rollout groups may still suffer from PCP, causing their posterior rewards to concentrate and weakening the optimization signal that drives subsequent policy updates. CR complements UDS by augmenting concentrated posterior rewards with a length-aware penalty, thereby restoring a discriminative training signal within the affected groups and enabling the policy to learn more effectively from the UDS-selected samples. The improved policy can, in turn, yield more informative uncertainty estimates for subsequent UDS updates. This dynamic interaction may explain why combining UDS with CR achieves the best performance on all three benchmarks, improving over the baseline by up to 6.7%.

CR cold start and UDS scheduler Table 3 evaluates the CR cold start and the UDS annealing scheduler. Without the cold start, we apply CR with $(\lambda, p_\delta, p_\alpha) = (0.5, 0.8, 0.2)$ from the beginning of training. This reduces accuracy by 4.9, 8.6, and 7.9 percentage points on MMLU-Pro, TheoremQA, and WebInstruct, respectively, suggesting that early regularization can hinder learning. Removing the UDS scheduler and fixing $(\tau_1, \tau_2) = (0.3, 0.7)$ throughout training also lowers accuracy across all three benchmarks. Together, these results support our claim that concentration becomes increasingly significant as the training proceeds. Thus, delaying CR and gradually adjusting the UDS thresholds during early training provides a more effective way to mitigate concentration.

7 Conclusion

We investigate the reliability of posterior probability rewards for verifier-free RL and identify the Posterior Concentration Phenomenon (PCP), where rewards become less discriminative as reasoning traces grow longer. We characterize PCP empirically and provide a theoretical formulation connecting it to log-likelihood concentration. Guided by these findings, we propose RLCPR, which explicitly accounts for concentration through uncertainty-aware data sampling and selective length regularization. Experiments show that RLCPR outperforms RLPR on six of seven reasoning benchmarks while requiring fewer cumulative rollout tokens. These results highlight the value of concentration-aware training for improving both effectiveness and token efficiency in probability-based RL.

AI USE STATEMENT

We used AI tools to polish the manuscript's language and presentation. The theoretical ideas and concepts were developed by the authors, with AI assistance used to check proofs for potential errors and refine their exposition. AI agents also assisted with code implementation and figure creation. The authors retain full responsibility for the correctness, originality, and integrity of the work.

REFERENCES

Tyler Bell, Avinash Mudireddy, Ivan Johnson-Eversoll, Soura Dasgupta, and Raghu Mudumbai. Slaves to the law of large numbers: An asymptotic equipartition property for perplexity in generative language models. *arXiv preprint arXiv:2405.13798*, 2024.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. TheoremQA: A theorem-driven question answering dataset. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7889–7901, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.489. URL https://aclanthology.org/2023.emnlp-main.489/.

Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *Advances in Neural Information Processing Systems*, 38:57654–57689, 2026.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Tairan Fu, Javier Conde, Gonzalo Martínez, María Grandury, and Pedro Reviriego. Multiple choice questions: Reasoning makes large language models (llms) more self-confident even when they are wrong. *arXiv e-prints*, pp. arXiv–2501, 2025.

John Hewitt, Christopher D Manning, and Percy Liang. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 3414–3427, 2022.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *Advances in Neural Information Processing Systems*, 38:162239–162262, 2026.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

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

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam

Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 3843–3857. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/18abbeef8cfe9203fdf9053c9c4fe191-Paper-Conference.pdf.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Wei Liu, Siya Qi, Xinyu Wang, Chen Qian, Yali Du, and Yulan He. Nover: Incentive training for language models via verifier-free reinforcement learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7450–7469, 2025.

Xueguang Ma, Qian Liu, Dongfu Jiang, Ge Zhang, Zejun Ma, and Wenhu Chen. General-Reasoner: Advancing LLM reasoning across all domains. *arXiv:2505.14652*, 2025. URL https://arxiv.org/abs/2505.14652.

OpenAI. *Introducing GPT-4.1 in the API*, 2025. URL https://openai.com/index/gpt-4-1/. Accessed: 2026.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

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. URL https://arxiv.org/abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Yunhao Tang, Sid Wang, Lovish Madaan, and Rémi Munos. Beyond verifiable rewards: Scaling reinforcement learning for language models to unverifiable data. *arXiv preprint arXiv:2503.19618*, 2025.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965, 2023.

Yue Wan, Xiaowei Jia, and Xiang Lorraine Li. Unveiling confirmation bias in chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 3788–3804, 2025.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 95266–95290. Curran Associates, Inc., 2024. doi: 10.52202/079017-3018. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ad236edc564f3e3156e1b2feafb99a24-Paper-Datasets_and_Benchmarks_Track.pdf.

Robert Welch, Emir Konuk, and Kevin Smith. The cost of reasoning: Chain-of-thought induces overconfidence in vision-language models. *arXiv preprint arXiv:2603.16728*, 2026.

Juming Xiong, Weixin Liu, Kevin Guo, Congning Ni, Junchao Zhu, Chongyu Qu, Chao Yan, Katherine Brown, Avinash Baidya, Xiang Gao, et al. Cora: Confidence-rationale alignment for reliable chain-of-thought reasoning. *arXiv preprint arXiv:2606.14961*, 2026.

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

Chenghao Yang, Sida Li, and Ari Holtzman. Llm probability concentration: How alignment shrinks the generative horizon, 2026. URL https://arxiv.org/abs/2506.17871.

Dongkeun Yoon, Seungone Kim, Sohee Yang, Sunkyoung Kim, Soyeon Kim, Yongil Kim, Eunbi Choi, Yireun Kim, and Minjoon Seo. Reasoning models better express their confidence. *Advances in Neural Information Processing Systems*, 38:103869–103896, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Tianyu Yu, Bo Ji, Shouli Wang, Shu Yao, Zefan Wang, Ganqu Cui, Lifan Yuan, Ning Ding, Yuan Yao, Zhiyuan Liu, et al. Rlpr: Extrapolating rlvr to general domains without verifiers. *arXiv preprint arXiv:2506.18254*, 2025.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Xiangxin Zhou, Zichen Liu, Anya Sims, Haonan Wang, Tianyu Pang, Chongxuan Li, Liang Wang, Min Lin, and Chao Du. Reinforcing general reasoning without verifiers. *arXiv preprint arXiv:2505.21493*, 2025.

APPENDIX

A PROOF OF LEM. 4.1

Since token-level entropy is non-negative, $\bar{h}_{x,\theta}(n) \geq 0$. By assumption, $\bar{h}_{x,\theta}(n)$ is eventually non-increasing, and therefore converges to some finite limit

$$\bar{h}_{x,\theta}(n) \to \bar{h}^{\infty}_{x,\theta} \geq 0. \tag{A1}$$

A finite prefix does not affect the asymptotic result, so in the following we consider sufficiently large n such that $\bar{h}_{x,\theta}(n)$ is non-increasing.

By definition,

$$\begin{aligned} h_{x,\theta}(n) &= n\bar{h}_{x,\theta}(n) - (n-1)\bar{h}_{x,\theta}(n-1) \\ &= \bar{h}_{x,\theta}(n) + (n-1)\big(\bar{h}_{x,\theta}(n) - \bar{h}_{x,\theta}(n-1)\big). \end{aligned} \tag{A2}$$

Hence,

$$0 \leq h_{x,\theta}(n) \leq \bar{h}_{x,\theta}(n) \tag{A3}$$

for all sufficiently large n.

We next show that the token-level entropy approaches $\bar{h}^{\infty}_{x,\theta}$ in the Cesàro L^1 sense. Since $h_{x,\theta}(n) \leq \bar{h}_{x,\theta}(n)$,

$$\begin{aligned} &\frac{1}{N}\sum_{n=1}^{N}\left(\bar{h}_{x,\theta}(n) - h_{x,\theta}(n)\right) \\ &\quad = \frac{1}{N}\sum_{n=1}^{N}\bar{h}_{x,\theta}(n) - \bar{h}_{x,\theta}(N) \to 0, \end{aligned} \tag{A4}$$

where the last step follows from $\bar{h}_{x,\theta}(n) \to \bar{h}^{\infty}_{x,\theta}$ and Cesàro convergence. Moreover, $|\bar{h}_{x,\theta}(n) - \bar{h}^{\infty}_{x,\theta}| \to 0$. Therefore,

$$\left|h_{x,\theta}(n) - \bar{h}^{\infty}_{x,\theta}\right| \leq \bar{h}_{x,\theta}(n) - h_{x,\theta}(n) + \left|\bar{h}_{x,\theta}(n) - \bar{h}^{\infty}_{x,\theta}\right|, \tag{A5}$$

which gives

$$\lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}\left|h_{x,\theta}(n) - \bar{h}^{\infty}_{x,\theta}\right| = 0. \tag{A6}$$

Recall that $W_n = \sum_{j=1}^{m} h_{x,\theta}(n+j)$. For fixed m and s,

$$\begin{aligned} |W_{n+s} - W_n| &\leq \sum_{j=1}^{m}|h_{x,\theta}(n+s+j) - h_{x,\theta}(n+j)| \\ &\leq \sum_{j=1}^{m}\Big(\left|h_{x,\theta}(n+s+j) - \bar{h}^{\infty}_{x,\theta}\right| + \left|h_{x,\theta}(n+j) - \bar{h}^{\infty}_{x,\theta}\right|\Big). \end{aligned} \tag{A7}$$

A fixed index shift preserves the Cesàro limit in Eq. (A6). Thus,

$$\lim_{N\to\infty}\frac{1}{N}\sum_{n=1}^{N}|W_{n+s} - W_n| = 0. \tag{A8}$$

Finally, consider an auxiliary random index $n \sim \mathrm{Unif}(1,\ldots,N)$. By Markov's inequality,

$$\mathbb{P}_{n\sim\mathrm{Unif}(1,\ldots,N)}\left(|W_{n+s} - W_n| > \epsilon\right) \leq \frac{1}{\epsilon N}\sum_{n=1}^{N}|W_{n+s} - W_n| \to 0, \tag{A9}$$

where the last step follows from Eq. (A8). The uniform distribution here is only an auxiliary device for measuring the fraction of reasoning lengths that violate the stability condition. Equivalently,

$$\frac{1}{N}\left|\left\{n \in 1,\ldots,N : |W_{n+s} - W_n| > \epsilon\right\}\right| = \mathbb{P}_{n\sim\mathrm{Unif}(1,\ldots,N)}\left(|W_{n+s} - W_n| > \epsilon\right) \to 0. \tag{A10}$$

Thus, the reasoning lengths for which $|W_{n+s} - W_n| > \epsilon$ form an asymptotically zero-density set. This completes the proof.

B Proof of Prop. 4.1

Recall from Eq. (11) that, for reasoning sequences of lengths $n+s$ and n,

$$\log\frac{\tilde{P}(y^* \mid x, Z_{n+s};\theta)}{\tilde{P}(y^* \mid x, Z_n;\theta)} = W_n - W_{n+s} + \rho_{n+s}(Z_{n+s}) - \rho_n(Z_n). \tag{A11}$$

Let $L_N \sim \mathrm{Unif}(\{1,\ldots,N\})$. Substituting $n = L_N$ into Eq. (A11), for any $\epsilon > 0$, the triangle inequality and union bound give

$$\begin{aligned} &\mathbb{P}\left(\left|\log\frac{\tilde{P}(y^* \mid x, Z_{L_N+s};\theta)}{\tilde{P}(y^* \mid x, Z_{L_N};\theta)}\right| > \epsilon\right) \\ &\quad \leq \mathbb{P}\left(|W_{L_N} - W_{L_N+s}| > \frac{\epsilon}{2}\right) + \mathbb{P}\left(|\rho_{L_N+s}(Z_{L_N+s}) - \rho_{L_N}(Z_{L_N})| > \frac{\epsilon}{2}\right). \end{aligned} \tag{A12}$$

We consider the two terms separately. By Lem. 4.1, the set of reasoning lengths satisfying $|W_{n+s} - W_n| > \epsilon/2$ has asymptotically zero density. Since L_N is uniformly distributed over $\{1, \ldots, N\}$,

$$\begin{aligned} &\mathbb{P}\left(|W_{L_N} - W_{L_N+s}| > \frac{\epsilon}{2}\right) \\ &\quad = \frac{1}{N}\left|\left\{n \in \{1, \ldots, N\} : |W_{n+s} - W_n| > \frac{\epsilon}{2}\right\}\right| \to 0. \end{aligned} \tag{A13}$$

For the residual term, define

$$p_n(\epsilon) := \mathbb{P}\left(|\rho_{n+s}(Z_{n+s}) - \rho_n(Z_n)| > \frac{\epsilon}{2}\right). \tag{A14}$$

By assumption, $p_n(\epsilon) \to 0$ as $n \to \infty$. Averaging over the auxiliary length L_N gives

$$\mathbb{P}\left(|\rho_{L_N+s}(Z_{L_N+s}) - \rho_{L_N}(Z_{L_N})| > \frac{\epsilon}{2}\right) = \frac{1}{N}\sum_{n=1}^{N} p_n(\epsilon) \to 0, \tag{A15}$$

where the last step follows from Cesàro convergence.

Combining Eqs. (A13) and (A15) with Eq. (A12), we obtain

$$\lim_{N\to\infty} \mathbb{P}\left(\left|\log \frac{\tilde{P}(y^* \mid x, Z_{L_N+s}; \theta)}{\tilde{P}(y^* \mid x, Z_{L_N}; \theta)}\right| > \epsilon\right) = 0. \tag{A16}$$

This completes the proof.

C Diagnostic Experiment: Relation between Uncertainty and Concentration

We investigate whether direct-answer uncertainty in Eq. (14) can identify examples that are prone to posterior concentration. We divide the training examples into two groups according to the binary indicator of concentration in Eq. (17) and compute the average direct-answer uncertainty in each group. Fig. A1 shows that, along the training process, questions whose resulting rollout groups satisfy the PCP condition consistently exhibit higher average uncertainty than those that do not. Thus, pre-reasoning uncertainty is associated with subsequent reward concentration and can serve as a practical proxy for concentration risk. This finding motivates uncertainty-aware data sampling: preferentially retaining lower-uncertainty examples can reduce concentration-prone rollouts and preserve more informative posterior-reward differences. Appx. E.2 provides the experimental details of Fig. A1, and Appx. F contains ablation study about the effect of different filtering rates.

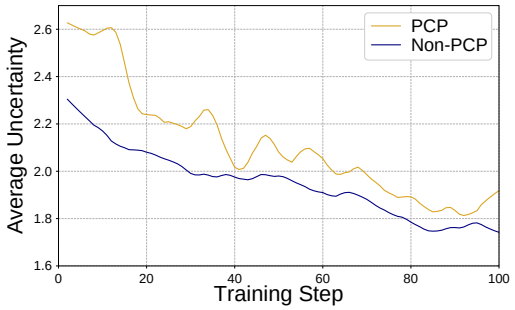


Figure A1: **Relation between uncertainty and concentration.** Throughout training, questions whose rollout groups satisfy the PCP condition consistently exhibit higher average direct-answer uncertainty than non-PCP questions. This finding motivates uncertainty-aware data sampling.

D Complexity Analysis

We compare the generation cost of uncertainty-aware data sampling (UDS) with post-generation reward filtering. Let $L = |x|$ denote the prompt length, n the reasoning length, and k the number of tokens used to estimate direct-answer uncertainty. We use an attention-cost approximation with KV caching, suppressing factors associated with model size. Generating m tokens after a prompt of length L then costs $\mathcal{O}(L^2 + Lm + m^2)$.

Uncertainty estimation. UDS estimates uncertainty using k direct-answer tokens and refreshes the cached estimates every f training steps. Amortizing each refresh over these steps gives the per-question estimation cost

$$C_{\text{unc}} = \mathcal{O}\left(\frac{L^2 + Lk + k^2}{f}\right). \tag{A17}$$

This amortization assumes that the refresh cost is distributed across the questions processed during the corresponding interval.

Expected rollout cost. The expected fraction of questions retained by UDS is

$$r = \sum_{i=1}^{3}(1 - \omega_i)(\tau_i - \tau_{i-1}), \qquad \tau_0 = 0, \quad \tau_3 = 1. \tag{A18}$$

For a fixed rollout group size, the expected generation cost per candidate question is therefore

$$C_{\text{UDS}} = \mathcal{O}\left(\frac{L^2 + Lk + k^2}{f} + r(L^2 + Ln + n^2)\right). \tag{A19}$$

In contrast, post-generation reward filtering generates rollouts for every candidate question before deciding which groups to discard, incurring

$$C_{\text{post}} = \mathcal{O}(L^2 + Ln + n^2). \tag{A20}$$

Thus, UDS reduces the rollout-generation term by the retention factor r, at the expense of uncertainty estimation. Under this approximation, net savings arise when the amortized estimation overhead is smaller than the avoided generation cost. Our default thresholds and filtering rates give $r = 0.55$, so approximately 45% of candidate questions are filtered before rollout generation. This analysis concerns generation cost; it does not quantify end-to-end training time, including reward scoring and policy updates.

E Implementation Details

E.1 PCP's Exploration-Study Setup

We analyze posterior concentration using Qwen3-4B-Base on MMLU-Pro and GPQA-Diamond. For each dataset, we randomly sample 128 questions and generate eight responses per question, yielding 1,024 (x, y, z) instances. We use a maximum response length of 1,536 tokens.

For Figs. 2a and 2b, we partition the traces according to answer correctness, $\mathbb{1}\{y = y^*\}$. We then fit separate third-order polynomial regressions of the posterior reward $P(y^* \mid x, z)$ against reasoning length for the correct and incorrect traces and report 95% confidence intervals.

For Fig. 2c, we divide the 1,024 responses into 16 bins of width 100 tokens. The k-th bin contains traces satisfying $100(k-1) \le |z| < 100k$. We compute the posterior-reward variance within each bin and regress these values against the corresponding bin centers, $100k - 50$.

For Fig. 2d, we preserve the rollout-group structure used by GRPO. For each rollout group G, we compute its average reasoning length and within-group posterior-reward variance as

$$\bar{\ell}_G = \frac{1}{|G|}\sum_{z \in G}|z|, \qquad v_G = \mathrm{Var}_{z \in G}\left[P(y^*_G \mid x_G, z)\right].$$

We then partition the 128 rollout groups into bins of width 100 tokens according to $\bar{\ell}_G$. For each non-empty bin, we plot the average $\bar{\ell}_G$ against the average v_G over all groups in that bin and fit the displayed regression trend with a 95% confidence interval. This analysis directly measures how posterior-reward dispersion within GRPO groups changes with their average reasoning length.

E.2 Uncertainty–Concentration Study Setup

In Fig. A1, we examine the relationship between direct-answer uncertainty and posterior concentration using Qwen3-4B-Base. We disable the standard-deviation-based filtering used in RLPR so that

concentration-prone rollout groups remain observable. At each training step, we partition the sampled questions into PCP and non-PCP subsets according to whether their resulting rollout groups activate the binary PCP indicator in Eq. (17). We then compute the average direct-answer uncertainty within each subset. For this diagnostic experiment, we train the model for 100 steps with a batch size of 320 and a GRPO group size of 8. Lastly, we report the subset-wise uncertainty averages throughout training. Additional ablations on the uncertainty-dependent filtering rates are provided in Appx. Sec. F.

E.3 RLCPR: Implementation Details

This section provides additional implementation details for the main experiments in Sec. 6.2. Similar to Yu et al. (2025), the policy model in RLCPR generates 8 responses per question, using a learning rate of $1e-6$. The KL divergence term is removed by setting the KL coefficient to 0. The batch size is set to 768 with 4 policy updates per step. The training prompt is provided in Sec. E.4.

E.4 RLCPR training prompt

We provide the complete system prompt used in RLCPR training and validation:

RLCPR training prompt for Qwen3

```
You are an assistant that solves problems by reasoning step by step.
You must always output your reasoning process inside <think> </think> tags,
and then your final answer inside <answer> </answer> tags.
Do not output anything else outside these tags.

Example:

<think>
The cost per jar of silver polish is $2.20. Fixed avoidable cost is $8,800.
Number of jars = 8,800 / 2.20 = 4,000.
</think>
<answer> 4,000 jars</answer>

Now follow the same format to answer the user's question.
user: {question}
Assistant:
```

Specifically, {question} refers to the input question. To follow the requested output format, we remove the special tokens <think> and </think> from the Qwen3's vocabulary.

F Additional Ablation Study

CR gate. Table A1 compares concentration-gated regularization with applying the length penalty to all groups. Removing the gate reduces accuracy by 1.6, and 3.5 percentage points on MMLU-Pro and TheoremQA, respectively, while obtaining only comparable points on WebInstruct. These results support selectively penalizing groups exhibiting posterior concentration: applying the penalty indiscriminately may discourage useful reasoning in groups whose posterior rewards remain informative.

Table A1: Ablation of the CR gate.

Setting	MMLU-Pro	TheoremQA	WebInst.
w/o CR gate	59.7	57.6	79.0
Full model	**61.3**	**61.1**	**78.9**

Table A2: Ablation of UDS sampling settings.

Cutoff (τ_1, τ_2)	Filter rates $(\omega_1, \omega_2, \omega_3)$	MATH-500	Minerva	AIME24
$(0.3, 0.7)$	$(0.7, 0.45, 0.2)$	67.5	47.7	10.2
$(0.3, 0.7)$	$(0.2, 0.45, 0.7)$	82.9	64.5	16.3
$(0.1, 0.9)$	$(0.2, 0.45, 0.7)$	81.0	58.4	12.9

Uncertainty distribution. Table A2 demonstrates the ablation study about changing the hyper-parameters of uncertainty-aware data sampling (UDS). Specifically, in the first two rows, we reverse the order of the filter rates. These results show that filtering more high-uncertainty

Table A3: Ablation of CR strength.

Reg. strength	MMLU-Pro	TheoremQA	WebInst.
$\lambda = 0.2$	58.4	55.2	73.9
$\lambda = 0.5$	61.3	61.1	78.9
$\lambda = 1.0$	59.5	59.5	76.2

data provides substantial improvement over filtering more low-uncertainty data, echoing our diagnostic experiment in Appx. C. In the last row, we increase the cutoff range, where fewer data are considered with particularly high or low uncertainty. In other words, most of the data are placed in the same bin, and the UDS does not essentially change the data distribution. This experiment also reduces the accuracy across all three mathematical benchmarks. It shows that reshaping the data distribution with direct-answer uncertainty effectively boosts the optimization performance.

Regularization strength. Table A3 shows the ablation study on the strengths of the concentration-aware regularization. Specifically, we change the weight of the regularization reward, *i.e.* λ. Overall, we recommend an optimal weight around 0.5, while a particularly high or low value of λ typically results in model degradation, possibly because of over or insufficient regularization. Notably, we fix the percentile levels to $(p_\delta, p_\alpha) = (0.8, 0.2)$ throughout our experiments without hyperparameter tuning. The reported gains thus demonstrate the effectiveness of this default configuration, while model- or benchmark-specific tuning may further improve performance.

G Discussion about PCP's Residual Condition

Our PCP analysis explicitly adopts convergence of the cross-trace residual difference as a sufficient condition. This assumption is motivated by Bell et al. (2024) and Yang et al. (2026), who together establish convergence of the length-normalized difference between negative log-likelihood and realized entropy, alongside empirical evidence of diminishing deviations and variability as generation length increases. Connecting these findings to posterior rewards offers a useful perspective on verifier-free RL, although establishing when the required residual differences converge remains an open theoretical question beyond our scope. Our contribution is to characterize conditions under which posterior rewards become less discriminative across reasoning traces and examine their implications for policy optimization. Guided by this perspective, RLCPR accounts for concentration through uncertainty-aware sampling and selective regularization to improve training efficiency and performance. Future work may strengthen this foundation by deriving sufficient conditions and finite-length bounds for residual-difference convergence.