QLPO: Quadrant-weighted Sampling for Length-aware Policy Optimization

Siwei Chen[1,2] **SiQi Chen**[3] **Xupeng Miao**[1,2] **Bin Cui**[1,2,4]

[1]School of Computer Science, Peking University
[2]Beijing Key Laboratory of Software and Hardware Cooperative Artificial Intelligence Systems, Peking University
[3]Department of Electronic Engineering, Tsinghua University
[4]Institute of Computational Social Science, Peking University (Qingdao)

Abstract

Recent large reasoning models often develop long chain-of-thought responses during reinforcement learning (RL), resulting in high inference latency and deployment cost. Existing methods for response length control typically rely on explicit length penalties or additional control modules, which require careful tuning and may compromise reasoning quality. We propose Quadrant-weighted sampling for Length-aware Policy Optimization (QLPO), a simple resampling-based variant of GRPO that introduces implicit length control without modifying the reward function. QLPO first over-generates candidate responses and then resamples the training group by preserving the empirical correct/incorrect ratio while favoring short correct responses and long incorrect responses. This reshapes the training distribution and implicitly encourages shorter model outputs. Across models ranging from 1.5B to 32B parameters, including both base models and strong reasoning models, QLPO consistently improves the accuracy–length trade-off. It reduces response length by 30% to 70% while preserving reasoning performance. These results suggest that structured resampling provides an effective and robust approach to efficient reasoning.

1 Introduction

The integration of Reinforcement Learning (RL) into Large Language Models has catalyzed the emergence of Large Reasoning Models (LRMs), such as OpenAI o-series (Jaech et al., 2024; OpenAI, 2025), Gemini 3 (Google, 2025), Claude 4.5 (Anthropic, 2025b;a), Grok 4 (xAI, 2025), DeepSeek-R1 (Guo et al., 2025), Qwen3 (Yang et al., 2025), Kimi-K2 (Team et al., 2025a). RL methods such as GRPO (Shao et al., 2024) promote long Chain-of-Thought reasoning, improving self-reflection and allowing LRMs to handle complex mathematical, coding and agentic tasks (Luo et al., 2025; Team et al., 2025b; Feng et al., 2025). However, this unbounded optimization paradigm has introduced a severe byproduct: the length explosion phenomenon. Driven by a reward signal that exclusively favors the final correct answer, LRMs consistently develop excessively verbose reasoning paths, generating thousands of redundant tokens. This verbosity not only inflates inference costs to prohibitive levels, but also causes severe deployment latency, creating a critical bottleneck for real-world applications.

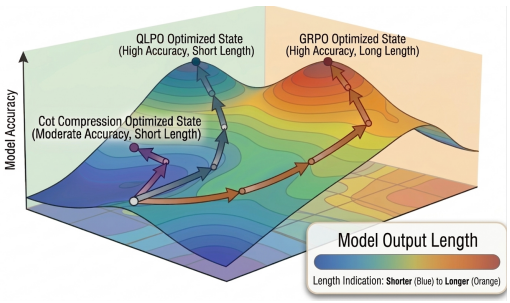

Figure 1: Conceptual illustration of the optimization path of different methods in the accuracy-length landscape.

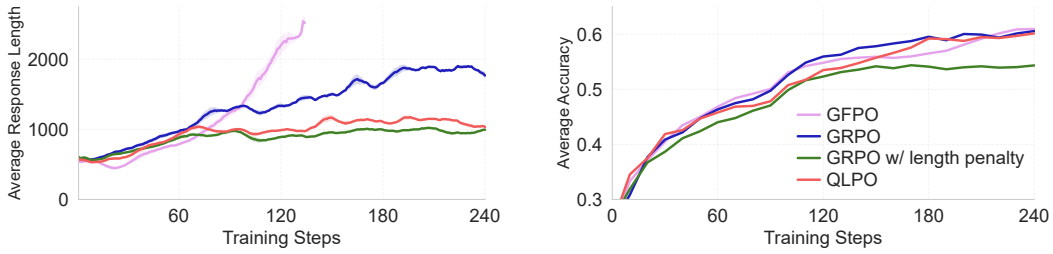

Figure 2: Training dynamics of different methods for Qwen2.5-32B on DAPO-MATH.

To enforce conciseness, existing works predominantly rely on explicit length penalties or complex step-level controllers. However, these methods suffer from complex designs with delicate hyperparameters that are difficult to tune. A bad coefficient easily distorts the RL objective, leading to a sharp performance drop on harder problems that require deep and extensive deductive steps. Other heuristic resampling approaches, such as Group Filtered Policy Optimization (GFPO) (Shrivastava et al., 2026), often face generalization challenges. Although selecting the shortest responses can be effective in certain scenarios, this strategy ignores the negative gradient contribution of incorrect samples and may even backfire, resulting in worse inference efficiency. Figure 2 illustrates one such failure case, whose underlying mechanism is discussed in detail in Sec. 3.3.

To alleviate the length explosion problem while overcoming the aforementioned limitations, we shift the paradigm from explicit reward hacking and empirical filtering to structured data resampling. In this paper, we propose **QLPO** (**Q**uadrant-weighted sampling for **L**ength-aware **P**olicy **O**ptimization), a simple and robust RL algorithm. Rather than explicitly penalizing long trajectories in the reward function, QLPO introduces Quadrant-weighted sampling after the RL rollout phase. Specifically, QLPO dynamically classifies the generated trajectories into four quadrants based on their correctness and relative length. By purposefully sampling specific trajectories before computing the policy gradient, QLPO naturally coerces the model to favor concise reasoning while protecting the model's capacity on complex tasks. We conduct a comprehensive evaluation of QLPO, demonstrating its effectiveness and robustness. Our primary contributions are summarized as follows:

- We introduce **QLPO**, a sampling-based variant of GRPO that achieves length-aware policy optimization while leaving the reward function, advantage estimator, and downstream GRPO optimization unchanged.
- We introduce a **group–trajectory gradient analysis** for group-based RL algorithms, which decomposes optimization into group-level gradient allocation induced by relative reward statistics and trajectory-level token-wise gradient realization. Under this view, QLPO is not a reward-shaping method but a selector that changes group composition and thereby implicitly encourages shorter outputs while preserving the main correctness signal.
- We show empirically that this simple modification consistently improves the accuracy–length frontier across a broad range of models, from 1.5B to 32B parameters and across both base and strong reasoning models. QLPO achieves substantial reductions in response length while maintaining competitive performance on challenging mathematical and scientific benchmarks.

2 Preliminaries

Group Relative Policy Optimization (Shao et al., 2024) is a widely used reinforcement learning algorithm that removes the need for a critic by estimating advantages from grouped samples (Schulman et al., 2017). Let $x \sim \mathcal{D}$ denote a prompt. For each prompt, GRPO samples a group of M responses $\mathcal{G}(x) = \{y_1, \ldots, y_M\}$ with $y_i \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$, and computes

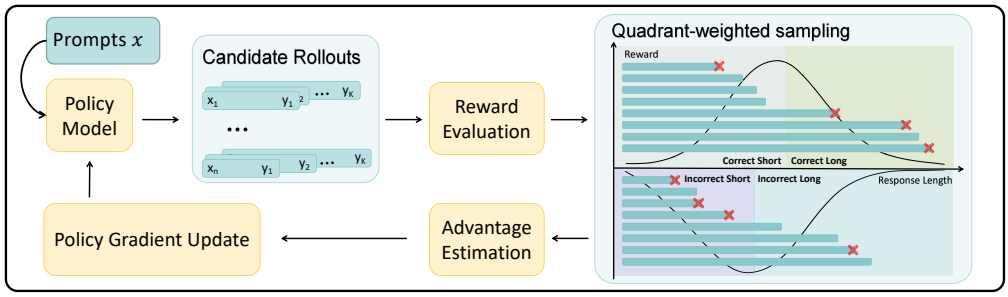

Figure 3: Overview of the QLPO training pipeline. QLPO first generates K candidate trajectories, partitions them according to correctness and relative length, and resamples M trajectories for standard group-relative advantage estimation and policy optimization.

the group-relative normalized advantage $A_i = \frac{r(x,y_i)-\bar{r}}{\sigma_r}$, where $\bar{r} = \frac{1}{M}\sum_{j=1}^{M} r(x,y_j)$ and $\sigma_r = \sqrt{\frac{1}{M}\sum_{j=1}^{M}(r(x,y_j)-\bar{r})^2}$.

The GRPO objective is given by

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_{x,\{y_i\}}\left[\frac{1}{\sum_{i=1}^{M}|y_i|}\sum_{i=1}^{M}\sum_{t=1}^{|y_i|}\min\Big(\rho_{i,t}(\theta)A_i,\ \text{clip}(\rho_{i,t}(\theta),\,1-\epsilon,\,1+\epsilon)\,A_i\Big)\right] - \beta\, D_{\text{KL}}\big(\pi_\theta \,\|\, \pi_{\theta_{\text{ref}}}\big)\,. \tag{1}$$

where $\rho_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t}|x,y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x,y_{i,<t})}$, and D_{KL} denotes the KL divergence between the current and ref policies. We adopt token-level loss normalization, which is the default choice in VeRL.

3 Quadrant-weighted sampling for Length-aware Policy Optimization

3.1 Overall pipeline

QLPO is a resampling-based variant of GRPO. For each prompt x, instead of directly applying GRPO to a group of size M, we first sample an over-generated candidate pool

$$\widetilde{\mathcal{G}}(x) = \{y_1, \ldots, y_K\}, \qquad K > M,$$

from the rollout policy $\pi_{\theta_{\text{old}}}$, and then construct a resampled group of size M for a standard GRPO update. We first partition the candidate pool by correctness, where $c(x,y) \in \{0,1\}$ is the correctness indicator: $\widetilde{\mathcal{G}}^+(x) = \{y \in \widetilde{\mathcal{G}}(x) : c(x,y) = 1\}$ and $\widetilde{\mathcal{G}}^-(x) = \{y \in \widetilde{\mathcal{G}}(x) : c(x,y) = 0\}$. Let $\widetilde{N}^+(x) = |\widetilde{\mathcal{G}}^+(x)|$ and $\widetilde{N}^-(x) = |\widetilde{\mathcal{G}}^-(x)|$, with $\widetilde{N}^+(x) + \widetilde{N}^-(x) = K$. We then set the target positive and negative counts in the final group as $N^+(x) = \text{round}\Big(M \cdot \frac{\widetilde{N}^+(x)}{K}\Big)$ and $N^-(x) = M - N^+(x)$, so that the final group contains exactly M samples while approximately preserving the empirical correct/incorrect ratio of the over-generated pool.

QLPO next partitions each correctness class into short and long responses within the same prompt. For each prompt x, we sort responses in $\widetilde{\mathcal{G}}^+(x)$ by length and split them into two halves, forming $\widetilde{\mathcal{G}}^+_{\text{short}}(x)$ and $\widetilde{\mathcal{G}}^+_{\text{long}}(x)$. We do the same for $\widetilde{\mathcal{G}}^-(x)$, obtaining $\widetilde{\mathcal{G}}^-_{\text{short}}(x)$ and $\widetilde{\mathcal{G}}^-_{\text{long}}(x)$. When a class has odd cardinality, the two subsets differ by at most one sample.

We then perform quadrant-weighted sampling to introduce a length preference. A length preference coefficient $\alpha \in (0,1)$ specifies the desired within-class retention ratio:

$$\text{Correct Long} : \text{Correct Short} = \alpha : 1, \qquad \text{Incorrect Short} : \text{Incorrect Long} = \alpha : 1.$$

Thus, when $\alpha < 1$, QLPO favors short-correct and long-incorrect responses. When $\alpha = 1$, this length preference disappears, reducing QLPO to GRPO. We prioritize the preferred quadrants, namely (Correct, Short) and (Incorrect, Long). So in practice, we use the ceiling function to determine the target numbers of samples drawn from these two quadrants, and obtain the sample counts for the remaining two quadrants by subtraction. This ensures that the final resampled group always contains exactly $N^+(x)$ correct samples and $N^-(x)$ incorrect samples. Accordingly, the target counts are calculated as

$$\widehat{N}^+_{\text{short}}(x) = \left\lceil \frac{1}{1+\alpha} N^+(x) \right\rceil, \qquad \widehat{N}^+_{\text{long}}(x) = N^+(x) - \widehat{N}^+_{\text{short}}(x),$$
$$\widehat{N}^-_{\text{long}}(x) = \left\lceil \frac{1}{1+\alpha} N^-(x) \right\rceil, \qquad \widehat{N}^-_{\text{short}}(x) = N^-(x) - \widehat{N}^-_{\text{long}}(x).$$

Finally, we sample trajectories uniformly within each quadrant to form the M responses for subsequent policy update. The general pipeline of QLPO is demonstrated in Figure 3.

3.2 A Group–Trajectory Gradient View of GRPO and QLPO

We propose a group- and trajectory-level gradient view to analyze group-based RL algorithms. The key idea is to decompose the update into two coupled levels: **group-level gradient allocation** and **trajectory-level token realization**. At the group level, the composition of the selected group determines how the gradient mass is distributed across trajectories through the group-relative advantages. At the trajectory level, this allocated signal is realized as token-wise gradients accumulated along each selected response. We follow the notation introduced in Section 2. Let $G(x) \subseteq \widetilde{\mathcal{G}}(x)$ denote the final selected group used for the GRPO update. $\bar{r}$ sets the center of the update, while $r(x, y_i) - \bar{r}$ determines whether a trajectory is reinforced or suppressed relative to the other samples in the same group.

Group-level gradient allocation. For a fixed prompt x and selected group $G(x)$, the GRPO gradient can be viewed as

$$\nabla_\theta \mathcal{L}_{\text{GRPO}}(\theta \mid x, G) = \sum_{y \in G(x)} A(y)\, \psi(y;\theta) - \beta \nabla_\theta D_{\text{KL}}(\pi_\theta \| \pi_{\theta_{\text{ref}}}),$$

where $A(y)$ is the group-relative advantage associated with trajectory $y \in G(x)$, and $\psi(y;\theta)$ denotes the trajectory-level gradient realization. For simplicity, we ignore the clipping-induced case distinction and use this decomposition only to expose the two-level structure of the update. Hence, the sampled group determines optimization through a group-level weighting rule: the average reward defines the baseline, and the normalized deviation from that baseline determines the sign and strength of each trajectory's contribution. Thus, GRPO allocates update strength according to each trajectory's relative position within the group: positive advantages reinforce a trajectory, negative advantages suppress it, and the magnitude of the advantage determines the strength of the effect.

Trajectory-level gradient realization. Let $\mathcal{T}(G)$ denote the set of valid generated tokens in $G(x)$, and let $N_G := |\mathcal{T}(G)|$. For a trajectory $y \in G(x)$, define its token-accumulated gradient contribution by

$$\psi(y;\theta) := \frac{1}{N_G} \sum_{t=1}^{|y|} \nabla_\theta \widetilde{\ell}_t(y;\theta),$$

where $\widetilde{\ell}_t(y;\theta)$ denotes the token-level clipped PPO term with the trajectory-level advantage factored out. Then the GRPO gradient for a fixed selected group can be written as

$$\nabla_\theta \mathcal{L}_{\text{GRPO}}(\theta \mid x, G) = \sum_{y \in G(x)} A(y) \frac{1}{N_G} \sum_{t=1}^{|y|} \nabla_\theta \widetilde{\ell}_t(y;\theta) - \beta \nabla_\theta D_{\text{KL}}(\pi_\theta \| \pi_{\theta_{\text{ref}}}).$$

This expression makes the two-level optimization structure explicit. At the group level, the group-relative advantage $A(y)$ determines the sign and relative magnitude of each

trajectory's contribution, and therefore sets the dominant correctness-oriented direction of the update. In this sense, it is the group-level advantage that carries the main signal associated with accuracy and reasoning quality. At the trajectory level, the token-wise gradients along y determine how this group-level signal is realized in parameter space through token-level accumulation. The resulting update is therefore jointly determined by group-level gradient allocation and trajectory-level gradient realization: the former governs the main performance-relevant direction of optimization, while the latter determines how that direction is concretely expressed through aggregated token gradients.

Under this view, QLPO introduces a selection preference over trajectories rather than modifying the reward or advantage itself. By preferentially retaining short-correct and long-incorrect trajectories, it changes which trajectories contribute gradient mass to the realized update, without directly perturbing the correctness signal encoded in the reward. This is importantly different from explicit length penalties: once a penalty term is added to the reward, the group-relative advantages are recomputed from the penalized rewards, which directly alters group-level gradient allocation. As a result, a long but correct trajectory that would have received a positive advantage under the original reward can become weakly positive or even negative after penalization. In this sense, length penalty does not merely discourage verbosity; it can change both the magnitude and even the sign of the optimization signal at the group level. By contrast, QLPO preserves the dominant correctness-oriented signal carried by the original group-relative advantage, while biasing the realized token-level update toward shorter successful responses and away from longer unsuccessful ones. Repeated over training, this induces an implicit preference for shorter outputs without directly corrupting the reward-defined objective.

3.3 GFPO from the Group–Trajectory Gradient View

GFPO (Shrivastava et al., 2026) is closely related to QLPO, but under the group–trajectory gradient view it has two structural weaknesses. First, a shortest-only selector does not preserve the correct/incorrect composition of the candidate pool, and thus perturbs the group-level gradient allocation. Second, shortest-only filtering does not guarantee shorter outputs. Under token-level loss normalization, the update depends on advantage-weighted token gradients aggregated over the selected group. For a selected group G, define the token-weighted average advantage $\bar{A}_{\text{tok}}(x; G) := \frac{1}{\sum_{y \in G} |y|} \sum_{y \in G} |y| \, A(y)$. Even in a relatively short-response region, $\bar{A}_{\text{tok}}(x; G)$ can be negative when negative-advantage trajectories contribute more token mass than positive-advantage ones. In that case, the update suppresses probability mass in the currently selected short region, causing training to drift away from it and worsening length explosion. We observe this behavior for GFPO in our setup (Figure 2). By contrast, QLPO preserves the group-level correctness signal and only reweights trajectories within each class, making it more reliable.

4 Experiments

4.1 Experimental Setup

Models and datasets. We evaluate QLPO on a diverse set of models, including Qwen2.5-3B, Qwen2.5-32B, Qwen3-30B-A3B, DeepScaleR-1.5B-Preview (DeepScaleR-1.5B) and DeepSeek-R1-Distill-Qwen-7B (DS-7B) (Yang et al., 2024; 2025). These models cover both dense and MoE architectures, as well as different levels of reasoning ability, allowing us to test whether QLPO generalizes across model families and capability regimes. We use MATH-lighteval (Hendrycks et al., 2021) for the 3B setting and DAPO-MATH (Yu et al., 2025) for the remaining models. In addition to text reasoning, we also conduct preliminary experiments in multimodal reasoning and code generation. Specifically, we evaluate QLPO for Qwen2.5-3B-VL (Bai et al., 2025) on Geo3K (Lu et al., 2021) and for Phi-3.5-mini-instruct (Abdin et al., 2024) on Eurus-2-RL-Data (Cui et al., 2025), to assess whether our method transfers beyond standard text-only reasoning settings.

Model	GSM8K		MATH-500		OlympiadBench		GPQA		AIME24	
	Acc.	Len.	Acc.	Len.	Acc.	Len.	Acc.	Len.	Acc.	Len.
DeepScaleR-1.5B	77.3	4,021	82.4	4,159	46.6	9,265	17.2	15,062	38.8	9,978
+ GRPO	85.2	3,434	87.2	4,095	50.7	17,138	28.2	27,758	40.8	9,674
+ QLPO	85.4	1,707	87.8	2,639	50.0	6,621	28.2	23,199	40.8	6,285
DS-7B	89.3	2,145	88.0	4,174	54.6	8,977	38.4	11,321	56.7	13,144
+ GRPO	91.2	2,388	92.1	4,119	58.5	8,213	44.4	19,321	57.1	12,237
+ QLPO	92.3	1,176	92.2	2,780	58.7	6,449	44.0	14,135	57.9	9,223
Qwen2.5-3B	76.5	291	56.2	1,444	20.5	4,000	28.8	1,798	4.5	4,798
+ GRPO	86.3	414	67.6	1,546	23.1	1,309	32.8	2,225	5.0	4,982
+ QLPO	84.8	300	69.4	1,013	26.1	885	31.7	721	5.0	2,437
Qwen2.5-32B	90.6	921	67.8	2,574	29.0	2,943	35.8	1,082	8.7	2,909
+ GRPO	96.0	341	87.6	862	45.2	1,397	54.0	1,455	28.3	1,880
+ QLPO	95.6	295	86.2	675	45.4	867	56.5	978	29.1	1,446
Qwen3-30B-A3B	96.0	1,833	96.2	5,448	88.2	10,219	56.1	6,064	78.3	14,167
+ GRPO	96.2	1,708	97.4	4,737	90.8	9,482	59.1	4,586	83.3	13,355
+ QLPO	96.1	1,215	96.8	3,528	89.3	7,070	60.6	4,016	82.1	9,692

Table 1: Accuracy–length trade-off across models under the base, GRPO, and QLPO settings.

Baselines. Our primary baseline is standard GRPO with an update-group size of $M = 8$. QLPO samples a candidate pool of $K = 16$ trajectories and selects $M = 8$ trajectories for the policy update. Thus, standard GRPO and QLPO use the same update-group size but different rollout budgets. To isolate the effect of quadrant-weighted selection from the effect of additional sampling, we further evaluate rollout-matched GRPO with $K = M = 16$, which generates the same number of trajectories as QLPO but updates the policy using all of them. We also compare against GFPO with $K = 16$ and $M = 8$.

We additionally compare against several open-source efficiency-oriented reasoning models, including L1, Laser, and ThinkPrune, using their publicly released checkpoints (Aggarwal & Welleck, 2025; Liu et al., 2026b; Fang et al., 2025; Hou et al., 2025).

Training Configuration and Hyperparameters. We implement QLPO based on VeRL framework (Sheng et al., 2025) with only about 200 lines of additional Python code. Unless otherwise specified, all methods are trained on 32 NVIDIA H20 GPUs with a global batch size of 128 using the Adam optimizer with a learning rate of 1×10^{-7}. For standard GRPO, we use a group size of $M = 8$. In QLPO, we first sample an over-generated candidate pool with size $K = 16$, apply quadrant-aware resampling, and then retain $M = 8$ responses for advantage estimation and policy updates. Therefore, QLPO has the same policy-update group size as standard GRPO, but uses a larger rollout candidate pool. The rollout-matched comparison in Section 5.1 controls for this difference in sampling budget.

Evaluation. We evaluate the trained models on five widely used reasoning benchmarks: MATH-500 (Lightman et al., 2023), GSM8K (Cobbe et al., 2021), OlympiadBench (He et al., 2024), GPQA (Rein et al., 2024) and AIME 2024. Unless otherwise specified, we sample model responses at temperature $T = 0.8$ with a maximum generation length of 32k tokens. We generate 16 responses per prompt for AIME24 and 3 responses per prompt for the remaining benchmarks. For GSM8K and MATH-500, correctness is determined by structured answer extraction followed by normalized exact matching. For the remaining benchmarks, we first apply rule-based matching; only when such matching fails do we fall back to GPT-4.1 as an auxiliary verifier. We report pass@1 accuracy and average output length, and use these metrics to evaluate the accuracy-efficiency trade-off of each method.

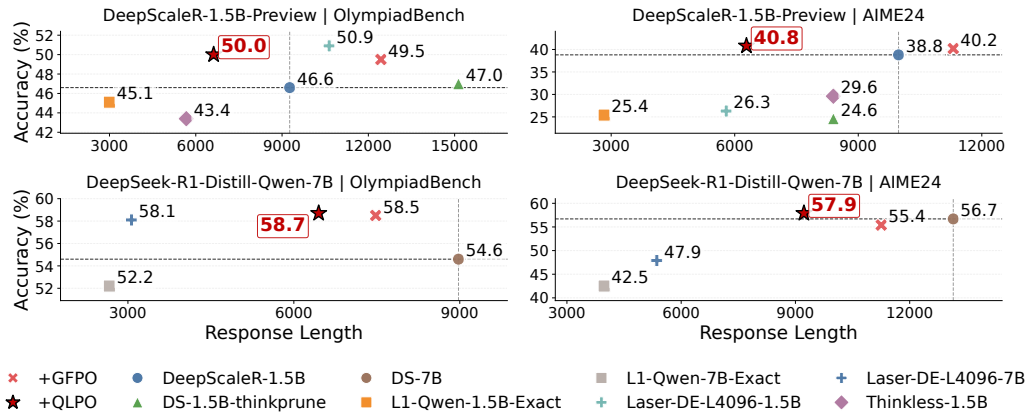

Figure 4: Comparison with representative compression baselines. Each point shows a different model, where lower response length and higher accuracy are preferred. Across both 1.5B and 7B settings, QLPO consistently achieves a more favorable trade-off, delivering substantial length reduction while preserving strong reasoning performance.

4.2 Main Results

QLPO Improves the Accuracy–Length Trade-off The magnitude of compression is substantial across a wide range of scales. For example, QLPO reduces OlympiadBench length from 17,138 to 6,621 for DeepScaleR-1.5B, AIME length from 12,237 to 9,223 for DeepSeek-R1-Distill-Qwen-7B and OlympiadBench length from 1,397 to 867 for Qwen2.5-32B. The largest relative reduction on these hard benchmarks reaches 67.6%, achieved on Qwen2.5-3B over GPQA (2,225 to 721), while the largest absolute reduction reaches 10,517 tokens on DeepScaleR-1.5B over OlympiadBench (17,138 to 6,621).

A central concern for QLPO is whether shorter responses come at the cost of reasoning performance on benchmarks that genuinely require extended multi-step reasoning. Our results show that this is not the case for QLPO. On the hardest benchmarks in Table 1, namely OlympiadBench, GPQA, and AIME24, QLPO matches or even slightly exceeds GRPO in most cases while producing substantially shorter outputs. Across the 15 model–benchmark pairs on these three benchmarks, the few observed drops are all small in magnitude and are well within the range of ordinary training stochasticity rather than evidence of systematic degradation. Overall, these results show that QLPO substantially improves response efficiency without sacrificing, and in some cases even improving, reasoning performance.

Accuracy–Length Trade-off Against Other Compression Methods Figure 4 compares QLPO with representative compression baselines. While some methods can shorten model outputs more aggressively, they typically do so by imposing a stronger compression bias, which may lead to noticeable degradation in reasoning performance, especially on more challenging benchmarks. By contrast, the main advantage of QLPO is not to maximize raw compression, but to improve reasoning efficiency while better preserving the performance gains of RL post-training. Empirically, in the 1.5B setting, QLPO achieves the strongest AIME24 performance (40.8) among the compared methods, while remaining substantially shorter than the original GRPO policy. On OlympiadBench, it maintains competitive accuracy with a much shorter response length than standard GRPO. In the 7B setting, QLPO again delivers the strongest AIME24 result (57.9) and OlympiadBench result (58.7) despite using fewer tokens than standard GRPO. Taken together, these results suggest that QLPO offers a more reliable accuracy–length trade-off than representative compression baselines.

4.3 QLPO Generalizes Across Diverse Tasks and Model Settings

Another notable result is the strong and consistent performance of QLPO across a range of text-based reasoning tasks. Across five models, QLPO stably reduces output length

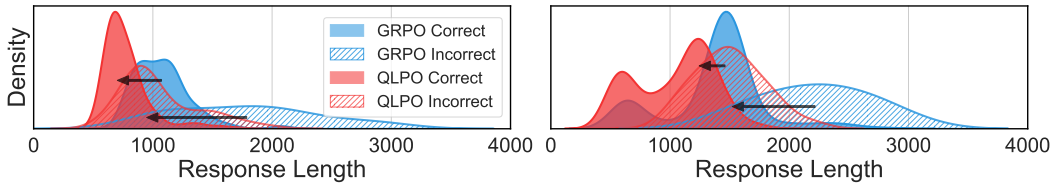

Figure 5: Response Length Distribution Shift Across Trajectory Correctness for DeepScaleR-1.5B-Preview on GSM8K (left) and DAPO-MATH (right). QLPO induces a larger leftward shift for incorrect trajectories.

while maintaining competitive reasoning performance, demonstrating its effectiveness in improving the accuracy–efficiency trade-off. Beyond standard text reasoning, we also conduct preliminary experiments in multimodal and code generation settings. Specifically, we evaluate QLPO on Qwen2.5-3B-VL with Geo3K and on Phi-3.5-mini-instruct with Eurus-2-RL-Data. In both cases, QLPO achieves clear length reduction while preserving task performance, with around 42% and 25% shorter outputs respectively. These results suggest that QLPO is not limited to text-only reasoning, but can generalize to broader reasoning settings, providing a simple and effective approach for improving inference efficiency. We demonstrate the detailed training dynamics in Appendix A.

4.4 Training–Inference Trade-off

The additional training overhead introduced by QLPO is limited in practice. The main source of extra cost lies in rollout, which is primarily memory-bound rather than compute-bound. Because decoding is autoregressive, throughput is constrained by sequential token generation and is often dominated by the long tail of exceptionally long responses. Consequently, increasing the number of rollouts does increase decoding cost, but the resulting wall-clock overhead is less than proportional to the increase in rollout count. Moreover, as training progresses, the model tends to produce shorter responses, which correspondingly reduces both rollout latency and policy-update time greatly. Empirically, QLPO incurs at most a 16.3% increase in wall-clock training time relative to GRPO, and can in some cases be up to 8.4% faster (Table 4). This trade-off is acceptable: a modest increase in training-time cost can be exchanged for substantial savings in inference-time computation.

5 Analysis

5.1 Rollout-Budget-Matched Comparison

QLPO uses a larger candidate pool than standard GRPO. To disentangle the benefit of quadrant-weighted selection from the benefit of observing additional trajectories, we introduce a rollout-budget-matched GRPO baseline. In this comparison, both QLPO and rollout-matched GRPO generate and evaluate $K = 16$ trajectories for each prompt. Rollout-matched GRPO uses all 16 trajectories for advantage estimation and policy updates, whereas QLPO selects $M = 8$ trajectories through quadrant-weighted sampling.

As shown in Table 2, simply increasing the GRPO group size from 8 to 16 does not improve response efficiency. On both models, rollout-matched GRPO produces slightly longer responses than standard GRPO and incurs substantially higher training cost. In contrast, QLPO observes the same 16-trajectory candidate pool while maintaining comparable accuracy and producing much shorter responses.

Relative to rollout-matched GRPO, QLPO reduces average response length by 45.1% on DeepScaleR-1.5B and 53.7% on Qwen2.5-3B. Moreover, because only eight selected trajectories enter advantage estimation and back-propagation, QLPO requires substantially less policy-update computation than 16-rollout GRPO. These results demonstrate that the accuracy–length gain does not arise from access to additional rollouts alone, but from the proposed structured selection mechanism.

Model	Method	Avg. Acc.	Avg. Len.	Train Time
DeepScaleR-1.5B	GRPO ($K = 8, M = 8$)	58.4	12,419	1.00
	GRPO ($K = 16, M = 16$)	58.2	14,734	2.13
	QLPO ($K = 16, M = 8$)	58.4	8,090	1.04
Qwen2.5-3B	GRPO ($K = 8, M = 8$)	43.0	2,095	1.00
	GRPO ($K = 16, M = 16$)	43.3	2,315	1.74
	QLPO ($K = 16, M = 8$)	43.4	1,071	1.13

Table 2: Rollout-budget-matched comparison. QLPO and 16-rollout GRPO generate and evaluate the same number of trajectories. QLPO performs policy updates only on the selected subset of eight trajectories, whereas rollout-matched GRPO updates on all 16. Accuracy and response length are averaged over the five main benchmarks. Training time is normalized by standard 8-rollout GRPO for each model.

5.2 Ablation Study

The effect of QLPO depends on both the candidate pool size K and the length preference coefficient α. We conduct an ablation study on DeepScaleR-1.5B-Preview. As shown in Figure 6, for the candidate pool size, we observe that QLPO requires sufficient over-generation to expose useful diversity among sampled responses. When $K = 1.5M$, the gain is limited, indicating that the sampled pool is too narrow for quadrant-aware resampling to identify concise high-quality trajectories. In contrast, $K = 2M$ already delivers strong and stable improvements, and further increasing K brings only marginal additional benefit. For the length preference coefficient, $\alpha = 1$ reduces QLPO to standard GRPO, since all correct responses are treated equally regardless of length. Empirically, we find that $\alpha = \frac{1}{3}$ works well and leads to a significant reduction in response length in our experiments. By contrast, $\alpha = 0$ corresponds to an extreme corner case that completely removes long but correct responses from the selected group, which is overly aggressive. In practice, this may discard informative long-horizon reasoning trajectories and lead to performance degradation.

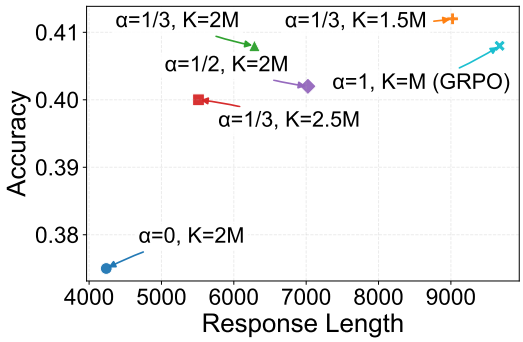

Figure 6: Ablation of hyperparameters for DeepScaleR-1.5B-Preview on DAPO-MATH.

5.3 Distribution Shift Analysis

QLPO improves reasoning efficiency by increasing the relative contribution of long incorrect responses, which makes it particularly effective at suppressing verbose yet unproductive reasoning trajectories. At the same time, QLPO still retains a portion of long correct responses, so the model's ability to learn useful long-horizon reasoning paths is preserved rather than removed. As shown in Figure 5, both correct and incorrect response distributions shift toward shorter lengths under QLPO, but the shift is visibly larger for incorrect responses. On GSM8K and DAPO-MATH, the average response length of incorrect trajectories decreases by 38%, while that of correct trajectories decreases by 23%.

Across questions of varying difficulty, we observe comparable relative reductions in response length. As shown in Figure 7, QLPO improves reasoning efficiency across different levels of problem difficulty. Quantitatively, the average length reduction is 30% to 35% on simpler benchmarks (GSM8K and MATH-500) and 33% to 38% on more complex benchmarks (OlympiadBench, GPQA and AIME24). This shows that QLPO promotes concise reasoning in a broad sense, rather than reflecting a compression effect confined to easier problems.

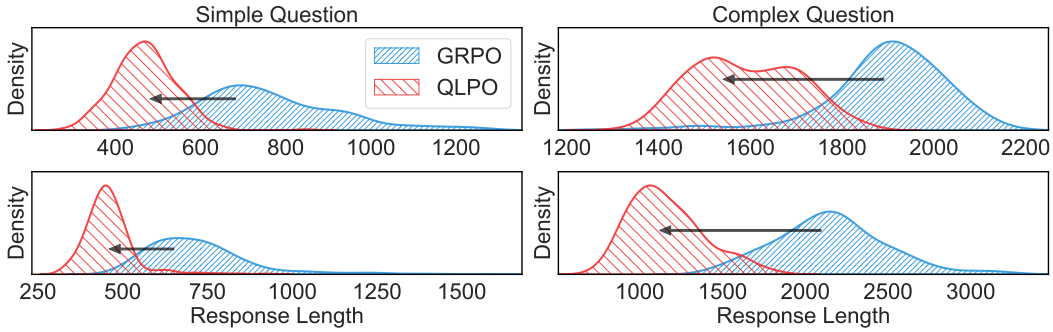

Figure 7: Response Length Distribution Shift Across Question Complexity for DeepScaleR-1.5B-Preview on GSM8K (left) and DAPO-MATH (right). QLPO shortens responses across both simple and complex questions, indicating a broad improvement in reasoning efficiency.

6 Related Work

RL-based Chain-of-Thought Compression. A large recent literature explicitly treats CoT length as a controllable resource and optimizes an accuracy–cost objective, typically combining a task reward with a penalty proportional to generated reasoning tokens. Representative sequence-level approaches include direct length-penalty RL objectives and reward shaping that encourage concise correct solutions while preserving performance on harder instances (Wang et al., 2026; Xiang et al., 2025; Yuan et al., 2025; Su & Cardie, 2025). Beyond global token penalties, several works introduce difficulty-adaptive schemes to realize "fast on easy, deep on hard" behavior (Shen et al., 2025; Liang et al., 2025; Xiang et al., 2025; Dumitru et al., 2025; Singhal et al., 2024). More structured control has also been explored via step/segment-level early-exit or pruning formulations that more closely resemble optimal stopping, including serial early-exit rewards, step-aware cost definitions, and redundancy-aware penalties (He et al., 2025; Wu et al., 2025; Dai et al., 2025; Hong et al., 2025). In parallel, instance-level mode selection frameworks learn when to trigger explicit CoT using RL-trained controllers or control tokens, enabling large efficiency gains by avoiding unnecessary deliberation on easy queries (Tu et al., 2025; Lou et al., 2025; Deng et al., 2025; Zhang et al., 2025; Liu et al., 2026a; Yu et al., 2026; Li et al., 2025b; Chen et al., 2025).

Sample Selection in LLM Training. Sample selection is increasingly important in LLM training, especially in RL-style post-training with multiple responses per prompt. Some methods explicitly filter sampled responses before optimization: GFPO retains a top-k subset to favor token-efficient learning, while DLER uses dynamic sampling and batch filtering to avoid degenerate reward groups (Shrivastava et al., 2026; Liu et al., 2025). Other work emphasizes that rollout-group construction and length-aware sampling can themselves shape optimization behavior, as seen in large-scale RL systems such as DAPO and in dynamic sampling methods such as LSPO (Yu et al., 2025; Chen et al., 2025). Related compression pipelines further employ rejection sampling or selective optimization to focus learning on more useful responses or token segments (Rakotonirina et al., 2026; He et al., 2025; Lou et al., 2025; Ha et al., 2025; Li et al., 2025a).

7 Conclusion

In this paper, we propose Quadrant-weighted sampling for Length-aware Policy Optimization (QLPO), a simple and effective reinforcement learning algorithm for mitigating length explosion in large reasoning models. By reshaping the training distribution through quadrant-weighted sampling, QLPO introduces an implicit preference for shorter reasoning without modifying the reward computation. Across a broad range of model scales and settings, QLPO consistently reduces response length by 30% to 70% while maintaining reasoning performance. Overall, QLPO provides a practical, robust, and easy-to-implement approach for improving the accuracy–efficiency trade-off of reasoning models.

Acknowledgment

This work is supported by National Natural Science Foundation of China (U23B2048, 62402011), Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (JYB2025XDXM108), High-performance Computing Platform of Peking University and The Fundamental Research Funds for the Central Universities. Xupeng Miao is the corresponding author.

References

Marah I. Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Hassan Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. In *Proceedings of the Conference on Language Modeling (COLM)*, 2025. URL https://openreview.net/forum?id=4jdIxXBNve.

Anthropic. Claude 4.5 opus. https://www.anthropic.com/news/claude-opus-4-5/, 2025a.

Anthropic. Claude 4.5 sonnet. https://www.anthropic.com/news/claude-sonnet-4-5/, 2025b.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Wei-Lin Chen, Liqian Peng, Tian Tan, Chao Zhao, Blake JianHang Chen, Ziqian Lin, Alec Go, and Yu Meng. Think deep, not just long: Measuring llm reasoning effort via deep-thinking tokens, 2026. URL https://arxiv.org/abs/2602.13517.

Weizhe Chen, Sven Koenig, and Bistra Dilkina. Lspo: Length-aware dynamic sampling for policy optimization in llm reasoning, 2025. URL https://arxiv.org/abs/2510.01459.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

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

Muzhi Dai, Chenxu Yang, and Qingyi Si. S-GRPO: Early exit via reinforcement learning in reasoning models. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=wNMK5o0Vfg.

Ken Deng, Zizheng Zhan, Wen Xiang, Wenqiang Zhu, Weihao Li, Jingxuan Xu, Tianhao Peng, Xinping Lei, Kun Wu, Yifan Yao, Haoyang Huang, Huaixi Tang, Kepeng Lei, Zhiyi Lai, Songwei Yu, Zongxian Feng, Zuchen Gao, Weihao Xie, Chenchen Zhang, Yanan Wu, Yuanxing Zhang, Lecheng Huang, Yuqun Zhang, Jie Liu, Zhaoxiang Zhang, Haotian Zhang, Bin Chen, and Jiaheng Liu. HiPO: Hybrid policy optimization for dynamic reasoning in LLMs, 2025. URL https://arxiv.org/abs/2509.23967.

Razvan-Gabriel Dumitru, Darius Peteleaza, Vikas Yadav, and Liangming Pan. Conciserl: Conciseness-guided reinforcement learning for efficient reasoning models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025. URL https://aclanthology.org/2025.findings-emnlp.927/.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: LLM learns when to think. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=ariVQf0KZx.

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*, 2025.

Google. Gemini 3 pro. https://deepmind.google/models/gemini/pro/, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Rui Ha, Chaozhuo Li, Rui Pu, and Sen Su. From "aha moments" to controllable thinking: Toward meta-cognitive reasoning in large reasoning models via decoupled reasoning and control, 2025. URL https://arxiv.org/abs/2508.04460.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. *CoRR*, abs/2402.14008, 2024. doi: 10.48550/ARXIV.2402.14008. URL https://doi.org/10.48550/arXiv.2402.14008.

Xingyang He, Xiao Ling, and Jie Liu. Smartthinker: Learning to compress and preserve reasoning by step-level length control, 2025. URL https://arxiv.org/abs/2507.04348.

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, 2021. URL https://arxiv.org/abs/2103.03874.

Jialiang Hong, Taihang Zhen, Kai Chen, Jiaheng Liu, Junlan Feng, Wenpeng Zhu, Jing Huo, Yang Gao, Depeng Wang, Haitao Wan, Xi Yang, Boyan Wang, Fanyu Meng, and Yuyao Zhang. Reconsidering overthinking: Penalizing internal and external redundancy in CoT reasoning, 2025. URL https://arxiv.org/abs/2508.02178.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *CoRR*, abs/2504.01296, 2025. doi: 10.48550/ARXIV.2504.01296. URL https://doi.org/10.48550/arXiv.2504.01296.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Gang Li, Yulei Qin, Xiaoyu Tan, Dingkang Yang, Yuchen Shi, Zihan Xu, Xiang Li, Xing Sun, and Ke Li. Rorecomp: Enhancing reasoning efficiency via rollout response recomposition in reinforcement learning. *arXiv preprint arXiv:2509.25958*, 2025a. URL https://arxiv.org/abs/2509.25958.

Yanhao Li, Lu Ma, Jiaran Zhang, Lexiang Tang, Wentao Zhang, and Guibo Luo. Leash: Adaptive length penalty and reward shaping for efficient large reasoning model. *arXiv preprint arXiv:2512.21540*, 2025b. URL https://arxiv.org/abs/2512.21540.

Tian Liang, Wenxiang Jiao, Zhiwei He, Jiahao Xu, Haitao Mi, and Dong Yu. Deepcompress: A dual reward strategy for dynamically exploring and compressing reasoning chains, 2025. URL https://arxiv.org/abs/2510.27419.

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. *CoRR*, abs/2305.20050, 2023. doi: 10.48550/ARXIV.2305.20050. URL https://doi.org/10.48550/arXiv.2305.20050.

Chang Liu, Yiran Zhao, Lawrence Liu, Yaoqi Ye, Csaba Szepesvári, and Lin F. Yang. Laconic: Length-aware constrained reinforcement learning for llm. *arXiv preprint arXiv:2602.14468*, 2026a. URL https://arxiv.org/abs/2602.14468.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. DLER: Doing length penalty right – incentivizing more intelligence per token via reinforcement learning, 2025. URL https://arxiv.org/abs/2510.15110.

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. Learn to reason efficiently with adaptive length-based reward shaping. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2026b. URL https://openreview.net/forum?id=hj9eKpqxQl. Poster.

Chenwei Lou, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. AdaCoT: Pareto-optimal adaptive chain-of-thought triggering via reinforcement learning, 2025. URL https://arxiv.org/abs/2505.11896.

Pan Lu, Sen Wang, Kaitao Long, Jian Zhou, and Jing Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4571–4584, 2021.

Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, Maurice Weber, et al. Deepcoder: A fully open-source 14b coder at o3-mini level. *Notion Blog*, 2025.

OpenAI. Openai o3. https://openai.com/index/introducing-o3-and-o4-mini/, 2025.

Nathanaël Carraz Rakotonirina, Ren Pang, Neha Anna John, Michael Bohlke-Schneider, and Momchil Hardalov. Correct, concise and complete: Multi-stage training for adaptive reasoning, 2026. URL https://arxiv.org/abs/2601.02972.

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. In *Proceedings of the Conference on Language Modeling (COLM)*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

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

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. DAST: Difficulty-adaptive slow-thinking for large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 2322–2331, Suzhou (China), 2025. Association for Computational Linguistics. ISBN 979-8-89176-333-3. doi: 10.18653/v1/2025.emnlp-industry.160. URL https://aclanthology.org/2025.emnlp-industry.160/.

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.

Vaishnavi Shrivastava, Ahmed Hassan Awadallah, Vidhisha Balachandran, Shivam Garg, Harkirat Behl, and Dimitris Papailiopoulos. Sample more to think less: Group filtered policy optimization for concise reasoning. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=UKOqoULbZS. Poster.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A long way to go: Investigating length correlations in rlhf. In *Proceedings of the Conference on Language Modeling (COLM)*, 2024. URL https://openreview.net/forum?id=G8LaO1P0xv.

Jinyan Su and Claire Cardie. Thinking fast and right: Balancing accuracy and reasoning length with adaptive rewards, 2025. URL https://arxiv.org/abs/2505.18298.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025a.

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*, 2025b.

Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. Learning when to think: Shaping adaptive reasoning in R1-style models via multi-stage RL. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Hs3FrjwyVZ.

Yujie Wang, Siwei Chen, Longzan Luo, Xinyi Liu, Xupeng Miao, Fangcheng Fu, and Bin Cui. Darts: Distribution-aware active rollout trajectory shaping for accelerating llm reinforcement learning. *arXiv preprint arXiv:2605.30859*, 2026.

Canhui Wu, Qiong Cao, Chang Li, Zhenfang Wang, Chao Xue, Yuwei Fan, Wei Xi, and Xiaodong He. Beyond token length: Step pruner for efficient and accurate reasoning in large language models, 2025. URL https://arxiv.org/abs/2510.03805.

xAI. Grok 4. https://x.ai/news/grok-4, 2025.

Violet Xiang, Chase Blagden, Rafael Rafailov, Nathan Lile, Sang Truong, Chelsea Finn, and Nick Haber. Just enough thinking: Efficient reasoning with adaptive length penalties reinforcement learning, 2025. URL https://arxiv.org/abs/2506.05256.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

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.

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. *arXiv preprint arXiv:2503.14476*, 2025.

Zewei Yu, Lirong Gao, Yuke Zhu, Bo Zheng, Sheng Guo, Haobo Wang, and Junbo Zhao. Stop unnecessary reflection: Training lrms for efficient reasoning with adaptive reflection and length coordinated penalty. *arXiv preprint arXiv:2602.12113*, 2026. URL https://arxiv.org/abs/2602.12113.

Danlong Yuan, Tian Xie, Shaohan Huang, Zhuocheng Gong, Huishuai Zhang, Chong Luo, Furu Wei, and Dongyan Zhao. Efficient RL training for reasoning models via length-aware optimization, 2025. URL https://arxiv.org/abs/2505.12284.

Yujian Zhang, Keyu Chen, Zhifeng Shen, Ruizhi Qiao, and Xing Sun. Adaptive dual reasoner: Large reasoning models can think efficiently by hybrid reasoning. In *NeurIPS 2025 Workshop on Efficient Reasoning*, 2025. URL https://arxiv.org/abs/2510.10207.

A Training Dynamics

Figure 8 and Figure 9 present the training curves of response length and accuracy in code-generation and multimodal settings, respectively. In both cases, QLPO shows the same qualitative pattern as in our main text experiments: compared with standard GRPO, it leads to shorter outputs over the course of training while preserving comparable task performance. The gap is especially clear in response length, whereas the accuracy curves remain close, suggesting that QLPO improves efficiency without introducing an obvious degradation in solution quality.

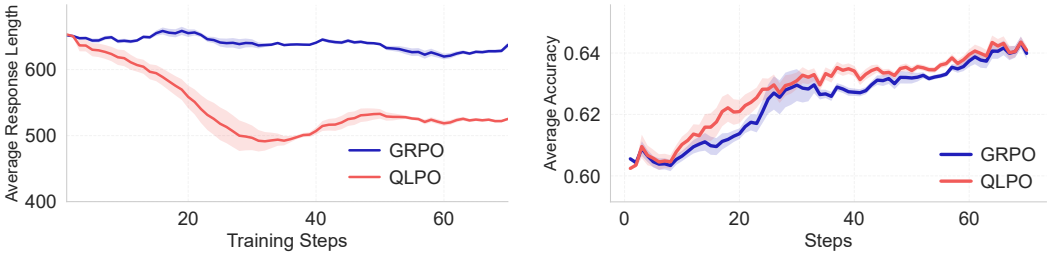

Figure 8: Phi-3.5-mini-instruct on Eurus-2-RL-Data.

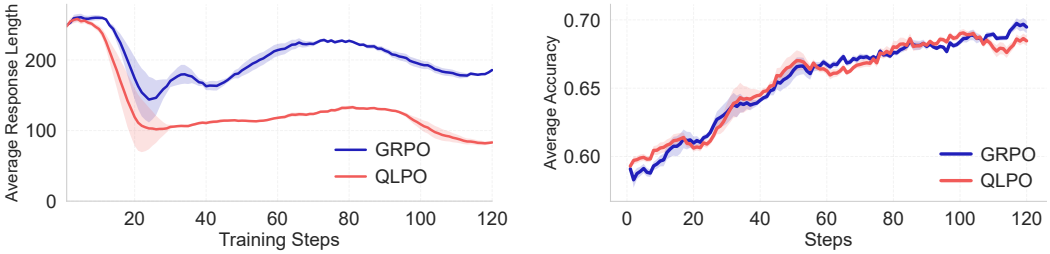

Figure 9: Qwen2.5-3B-VL on Geo3K.

B Why Can QLPO Produce Shorter Responses with Higher Accuracy?

To illustrate the qualitative difference induced by QLPO, we show a representative example from real model outputs below. The two responses are produced by Qwen2.5-32B models trained with GRPO and QLPO, respectively, on the prompt: "Solve for x: $4(x+2)-3=2x+9$. Show your reasoning step by step and verify your final answer."

> **GRPO response:** "To solve the equation $4(x+2)-3=2x+9$ for x, we follow these steps. First, distribute the 4 on the left-hand side to obtain $4x+8-3=2x+9$. Next, combine like terms on the left-hand side, which gives $4x+5=2x+9$. Then subtract $2x$ from both sides to isolate the x-terms on one side, yielding $2x+5=9$. After that, subtract 5 from both sides to get $2x=4$. Finally, divide both sides by 2, so $x=2$.
>
> To verify the solution, substitute $x=2$ back into the original equation: $4(2+2)-3=2(2)+9$. Simplifying gives $16-3=4+9$, and evaluating both sides yields $13=13$. Since both sides are equal, the solution is verified. Therefore, the solution is 2."
>
> **QLPO response:** "Distribute the 4 to get $4x+8-3=2x+9$, then simplify to $4x+5=2x+9$. Subtract $2x$ from both sides to obtain $2x+5=9$, subtract 5 to get $2x=4$, and divide by 2 to find $x=2$.
>
> To verify, substitute $x=2$ into the original equation: $4(2+2)-3=2(2)+9$. This simplifies to $16-3=4+9$, so $13=13$. Therefore, the solution is correct."

Compared with the GRPO response, the QLPO response preserves the same overall reasoning structure—expanding and simplifying the equation, isolating the variable, and verifying the solution—but presents it more concisely and with less redundant narration. This example illustrates that the shorter response under QLPO need not arise from truncating essential reasoning. Instead, the compression mainly comes from removing repeated restatements, excessive verbal scaffolding, and other low-value tokens, while retaining the intermediate computations necessary for a correct solution.

This observation is consistent with a broader interpretation of why shorter responses could also be effective. CoT length is not equivalent to reasoning capability: a long response may reflect genuine multi-step reasoning, but it may also contain redundant explanations, repeated verification, or low-value detours that increase token count without contributing meaningful computation. Recent work has made this distinction more explicit. For example, Chen et al. (2026) argue that surface-level token length is an unreliable proxy for reasoning quality, and suggests that effective reasoning is better characterized by "deep-thinking tokens," i.e., tokens associated with substantial internal belief revision across model layers. From this perspective, QLPO does not simply bias the model toward shorter outputs; rather, by reallocating gradient signal away from inefficient trajectories and toward more informative ones, it may encourage the model to produce a higher concentration of tokens that correspond to genuinely useful reasoning. In this sense, QLPO may compress the surface form of CoT without reducing the effective depth of reasoning, allowing the model to preserve or even strengthen the critical intermediate computations that support accurate answers. This provides a plausible explanation for why shorter generations under QLPO need not hurt performance and can sometimes even improve it.

C Stability Across Random Seeds

To assess training stability, we repeat training with three random seeds on two representative settings: DeepSeek-R1-Distill-Qwen-7B and Qwen2.5-3B. For each seed, we evaluate the final checkpoint on the same five benchmarks used in Table 1. Accuracy and response length are first averaged over the five benchmarks, after which we report the mean and standard deviation across training seeds.

Model	GRPO Acc.	QLPO Acc.	GRPO Len.	QLPO Len.
DS-7B	68.7 ± 0.5	69.0 ± 0.4	$9,256 \pm 1,076$	$6,753 \pm 830$
Qwen2.5-3B	43.0 ± 0.7	43.4 ± 0.6	$2,095 \pm 245$	$1,071 \pm 182$

Table 3: Mean and standard deviation across three independent training seeds. Accuracy and response length are averaged over the five main benchmarks for each seed.

Across both settings, the differences in average accuracy are small relative to seed-level variation, whereas the response-length reductions are substantially larger and consistent across seeds. QLPO reduces average length by 27.0% on DS-7B and by 48.9% on Qwen2.5-3B. These results indicate that the compression effect is not attributable to a single favorable random seed.

D Detailed Training-Time Results

Section 5.1 presents the controlled rollout-budget comparison on two representative models. Here, we report the normalized wall-clock training time of QLPO relative to standard 8-rollout GRPO across all five model settings.

Table 4 reports the normalized wall-clock training time of QLPO relative to standard 8-rollout GRPO across five models. Overall, QLPO introduces modest overhead in the evaluated single-turn reasoning settings. The normalized training time ranges from 0.916 to 1.163, corresponding to an 8.4% reduction to a 16.3% increase relative to standard GRPO. Although QLPO samples twice as many initial candidates, the wall-clock increase is substantially smaller than the nominal increase in rollout count. This is because only the selected subset enters advantage estimation and back-propagation, and because QLPO progressively shortens generated responses during training. In some settings, such as Qwen2.5-32B, these savings offset the additional candidate generation and result in a lower overall training time.

Model	QLPO time (GRPO = 1.0)
DeepScaleR-1.5B-Preview	1.035
DeepSeek-R1-Distill-Qwen-7B	1.107
Qwen2.5-3B	1.132
Qwen2.5-32B	0.916
Qwen3-30B-A3B	1.163

Table 4: Normalized wall-clock training time of QLPO relative to standard GRPO with $K = M = 8$. For each model, the training time of standard GRPO is normalized to 1.0.

E Full Experimental Configuration

Table 5 summarizes the training and evaluation configuration used in the main experiments.

Hyperparameter	Value
Maximum prompt length	2k tokens
Maximum response length	32k tokens
Prompt truncation	Left truncation
Base RL algorithm	GRPO
KL reward	Disabled
KL loss coefficient	0.01
Loss normalization	Token mean
Standard GRPO group size	8
QLPO candidate-pool size K	16
QLPO update-group size M	8
QLPO length coefficient α	$1/3$
Rollout-matched GRPO group size	16
Optimizer	Adam
Learning rate	1×10^{-7}
Warm-up steps	10
Weight decay	0.1
Gradient clipping	1.0
Prompt batch size	128
PPO mini-batch size	256
PPO clipping range	$[0.8, 1.2]$
Rollout temperature	1.0
Rollout top-p	1.0
Rollout top-k	-1
Evaluation temperature	0.8
Hardware	32 NVIDIA H20 GPUs
Parallelism	FSDP + vLLM
Tensor-parallel size	4
Offloading	Enabled
Evaluation interval	Every 10 steps
Checkpoint interval	Every 50 steps

Table 5: Full configuration for the main experiments.

The maximum response length is 32k tokens. No more than 0.1% of generated responses reach this limit during training and evaluation. Therefore, the observed reductions in response length are not primarily caused by output truncation.

F Limitations

While QLPO is an effective method for mitigating the length explosion problem in reasoning models, it has a few practical limitations. First, the method relies on an over-generation mechanism to construct the resampled training group. Although our experiments indicate that the training time often remains comparable to standard GRPO—since the model learns to generate shorter, faster-to-decode responses over time—sampling extra candidate

trajectories can still incur additional computational overhead and potentially result in slower overall training, especially under constrained hardware configurations. Second, QLPO is explicitly designed for scenarios that suffer from excessively verbose chain-of-thought reasoning. Consequently, for models that naturally produce concise outputs, or for tasks that inherently require short, fixed-length text generation (e.g., simple question-answering without reasoning traces or rigid formatting tasks), applying QLPO provides little meaningful benefit. In such cases where the output length is already naturally bounded, standard reinforcement learning methods remain the more appropriate choice.

G Future Directions

QLPO is based on a quadrant-weighted sampling strategy, which is simple and effective. Future work could explore more fine-grained allocation rules that distinguish responses not only by binary correctness and relative length, but also by richer attributes such as difficulty, confidence, or step-level reasoning quality. Another promising direction is to replace the fixed resampling heuristic with adaptive policies that learn how much training weight should be assigned to different types of trajectories. Such extensions may yield better accuracy-efficiency trade-offs. However, such extensions might introduce additional complexity and warrant further exploration.