
G²RPO-A: GUIDED GROUP RELATIVE POLICY OPTIMIZATION WITH ADAPTIVE GUIDANCE

Yongxin Guo^{1,2*}, Wenbo Deng^{1*}, Zhenglin Cheng³, Xiaoying Tang¹

¹School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen,

Guangdong, 518172, P.R. China ²Alibaba Group

³School of Engineering, Westlake University

August 19, 2025

ABSTRACT

Reinforcement Learning with Verifiable Rewards (RLVR) has markedly enhanced the reasoning abilities of large language models (LLMs). Its success, however, largely depends on strong base models with rich world knowledge, yielding only modest improvements for small-size language models (SLMs). To address this limitation, we investigate Guided GRPO, which injects ground-truth reasoning steps into roll-out trajectories to compensate for SLMs' inherent weaknesses. Through a comprehensive study of various guidance configurations, we find that naively adding guidance delivers limited gains. These insights motivate G²RPO-A, an adaptive algorithm that automatically adjusts guidance strength in response to the model's evolving training dynamics. Experiments on mathematical reasoning and code-generation benchmarks confirm that G²RPO-A substantially outperforms vanilla GRPO. Our code and models are available at <https://github.com/T-Lab-CUHKZ/G2RPO-A>.

1 Introduction

Recent advancements in reasoning-centric large language models (LLMs), exemplified by DeepSeek-R1 Guo et al. [2025], OpenAI-o1 Jaech et al. [2024], and Qwen3 Yang et al. [2025a], have significantly expanded the performance boundaries of LLMs, showcasing the immense potential of reasoning-enhanced models. Building upon robust base models with comprehensive world knowledge, these reasoning-focused LLMs have achieved breakthrough progress in complex domains such as mathematics Guan et al. [2025], coding Souza et al. [2025], HUANG et al. [2025], and other grounding tasks [Li et al., 2025b, Wei et al., 2025]. At the core of this success lies Reinforcement Learning with Verifiable Rewards (RLVR) [Shao et al., 2024, Chu et al., 2025, Liu et al., 2025]. This innovative approach, which employs reinforcement learning techniques in LLMs using rule-based outcome rewards, has garnered significant attention in the AI community. RLVR has demonstrated remarkable improvement in generalization across a wide spectrum of downstream tasks [Jia et al., 2025, Wu et al., 2025], positioning it as a pivotal advancement in the field of artificial intelligence.

As the de-facto algorithm, Group Relative Policy Optimization (GRPO) [Shao et al., 2024] improves upon Proximal Policy Optimization (PPO) [Schulman et al., 2017] by removing the need for a critic model through inner-group response comparison, thereby speeding up the training.

Capacity of small-size LLMs limit the performance gains of GRPO. Despite GRPO's success with large-scale LLMs, its effectiveness is significantly constrained when applied to smaller LLMs. Recent research [Ye et al., 2025, Muennighoff et al., 2025a] reveals that GRPO's performance gains highly depend on the base model's capacity Bae et al. [2025], Xu et al. [2025], Zhuang et al. [2025]. Consequently, small-scale LLMs (SLMs) show limited improvement under GRPO (Table 3, 8), exposing a critical scalability challenge in enhancing reasoning capabilities across diverse model sizes. To address this challenge, researchers have explored various approaches: distillation Guo et al. [2025], multi-stage training Xu et al. [2025] prior to RLVR, and selective sample filtering Xiong et al. [2025], Shi et al.

*Equal Contribution.

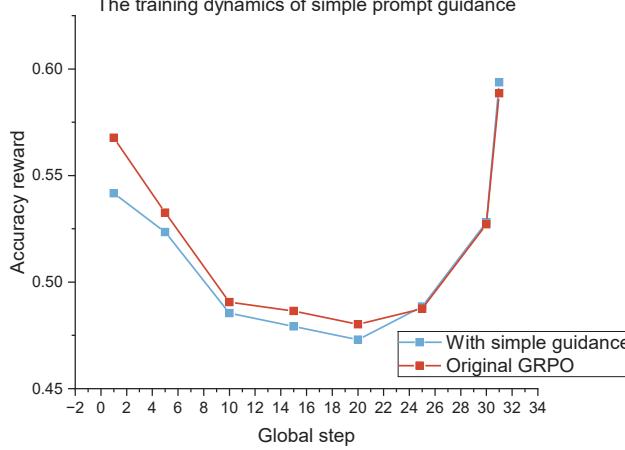


Figure 1: **Naive guidance does not help.** Using Qwen2.5-Math-7B as the base model, we train it on the s1K-1.1 dataset for a single epoch with a simple, fixed-length guidance (naive guidance). The naive guidance method shows a temporary increase in the accuracy reward during the early training stages, but it quickly becomes indistinguishable from the vanilla GRPO curve.

For the sequence of positive numbers: $\{a_n\}$, the sum of the first n terms is b_n , and the product of the first n terms of the sequence $\{b_n\}$ is c_n , and it is given that $b_n + 2c_n = 1$ ($n \in \mathbb{Z}_+$). Find the number in the sequence $\left\{\frac{1}{a_n}\right\}$ that is closest to 2013.



Guidance:

Alright, let's try to tackle this problem step by step. So, we have a sequence of positive numbers $\{a_n\}$. The sum of the first n terms is b_n . Then, the product of the first n terms of the sequence b_n is c_n . It's given that for every positive integer n , $b_n + 2c_n = 1$. We need to find the number in the sequence $\left\{\frac{1}{a_n}\right\}$ that is closest to 2013.



LLMs continue response:

Hmm, okay. Let me parse this again. The sum of the first terms of $\{a_n\}$ is b_n . Then, the product of the first terms of b_n is c_n . So, $c_n = b_1 * b_2 * \dots * b_n$. And for each n , $b_n + 2c_n = 1$. Interesting. So, for each n , both b_n and c_n are involved in this equation. Since $\{a_n\}$ are positive, all the b_n will be positive as well, and hence c_n is also positive.

Figure 2: **Illustration of roll-outs with guidance.** An example of using high-quality thinking trajectories to guide models.

[2025]. However, these methods precede RLVR or suffer performance degradation in complex problems (Table 7). Consequently, optimizing the RLVR process for efficient learning in SLMs remains an open challenge, representing a critical frontier in AI research.

Adaptive guidance as a solution. We propose incorporating guidance into the roll-out process to facilitate the generation of high-quality, reward-worthy candidates (Figure 2). However, our initial findings revealed that the implementation of simple fixed-length guidance to the prompts (naive guidance) failed to improve overall performance (Figure 1). Through a comprehensive analysis of the guidance mechanism—varying both the proportion of guided roll-outs within GRPO batches and the guidance length over training epochs—we obtained two key findings: (1) Code-generation tasks benefit from a higher guidance ratio than mathematical reasoning tasks, and smaller models likewise require more guidance than larger ones. (2) The optimal guidance length evolves throughout training and is highly context-dependent, rendering simple, predefined schedules ineffective. In response, we introduce the Guided Group Relative Policy Optimization with Adaptive Guidance (G²RPO-A) algorithm. This innovative approach dynamically adjusts guidance length based on the model’s real-time learning state, offering a sophisticated solution to the challenges of enhancing small-size LLMs’ performance in RLVR processes. The key contributions of this paper are summarized as follows:

- We enhance GRPO for small-scale LLMs by injecting guidance into the rollout thinking trajectories and conduct a systematic analysis of the effects of key guidance configurations, specifically focusing on the guidance ratio and guidance length.
- Our study also examines the importance of hard training samples. We find that integrating these samples into the dataset using a curriculum learning approach, and aided by the guidance mechanism, significantly boosts the training efficiency of our method for SLMs.

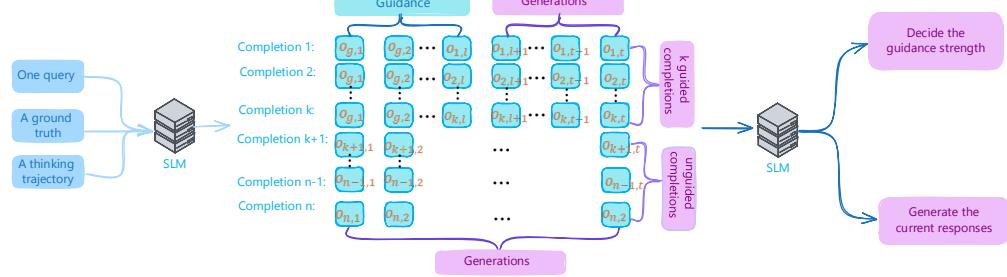


Figure 3: **Overview of G²RPO-A.** Each step we split roll-outs into a guided set and an unguided set. We then compare the current rewards with those from the previous steps; the resulting ratio determines the future guidance length.

- Drawing on these findings, we introduce G²RPO-A, an adaptive algorithm that automatically adjusts guidance length in response to the evolving training state. Our experimental results demonstrate the effectiveness of the proposed G²RPO-A algorithm.
- We evaluate our method on mathematical reasoning and coding tasks with several models—including the Qwen3 series, DeepSeek-Math-7B-Base, and DeepSeek-Coder-6.7B-Base—and observe substantial performance gains over both vanilla GRPO and simple guided baselines.

2 Related Works

The introduction of chain-of-thought (CoT) prompting has markedly improved LLM performance on complex reasoning tasks [Wei et al., 2022, Kojima et al., 2022]. Complementing this advance, Reinforcement Learning with Verifiable Rewards (RLVR) has emerged as a powerful training paradigm for reasoning-centric language models Yue et al. [2025], Lee et al. [2024]. The de-facto RLVR algorithm, Group Relative Policy Optimization (GRPO) [Guo et al., 2025] delivers strong gains on various benchmarks while remaining training-efficient because it dispenses with the need for a separate critic network Wen et al. [2025], Shao et al. [2024]. Recent efforts to improve GRPO have explored several directions. Some approaches focus on refining the core GRPO objective, either by pruning candidate completions Lin et al. [2025] or by removing normalization biases Liu et al. [2025]. Separately, another studies aim to enhance the training signal and stability. DAPO Yu et al. [2025], for instance, introduces dense, step-wise advantage signals and decouples the actor-critic training to mitigate reward sparsity.

However, adapting GRPO-style algorithms to small-scale LLMs remains challenging due to the sparse-reward problem Lu et al. [2024], Nguyen et al. [2024], Dang and Ngo [2025]. Recent studies have therefore focused on improved reward estimation Cui et al. [2025]. TreeRPO Yang et al. [2025b] uses a tree-structured sampling procedure to approximate step-wise expected rewards, and Hint-GRPO Huang et al. [2025] applies several reward-shaping techniques. Other lines of research investigate knowledge distillation [Guo et al., 2025], multi-stage pre-training before RLVR Xu et al. [2025], and selective sample filtering Xiong et al. [2025], Shi et al. [2025]. In our experiments, however, these filtering or sampling strategies are performed either only before RLVR or do not improve performance on more complex tasks. In this paper, we introduce a guidance mechanism that injects ground-truth reasoning steps directly into the model’s roll-out trajectories during RL training. Because the guidance is provided online, the proposed method can still learn effectively from difficult examples while mitigating the sparse-reward issue.

The role of guidance in GRPO-style training remains underexplored. Two concurrent studies have addressed related questions [Nath et al., 2025, Park et al., 2025], but both simply append guidance tokens to the input prompt, offering neither a systematic analysis of guidance configurations nor a mechanism that adapts to the changing training state of the model. We show that naive guidance often fails to improve performance because it yields low expected advantage. To remedy this, we provide a comprehensive examination of how guidance length, ratio, and scheduling affect learning, and we introduce G²RPO-A, an adaptive strategy that dynamically adjusts guidance strength throughout training.

3 Preliminary

Group Relative Policy Optimization (GRPO). Given a prompt, GRPO Shao et al. [2024] samples G completions and computes their rewards $\{r_i\}_{i=1}^G$. Define the t^{th} token of the i^{th} completion as $o_{i,t}$. GRPO then assigns a advantage, $\hat{A}_{i,t}$ to it. The optimization objective is defined as:

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \left[\min\left(w_{i,t} \hat{A}_{i,t}, \text{clip}(w_{i,t}, 1-\epsilon, 1+\epsilon) \hat{A}_{i,t}\right) - \beta \mathcal{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \right], \quad (1)$$

where the importance weight $w_{i,t}$ is given by

$$w_{i,t} = \frac{\pi_\theta(o_{i,t} | q, o_{i,<t})}{[\pi_{\theta_{\text{old}}}(o_{i,t} | q, o_{i,<t})]}. \quad (2)$$

The clipping threshold ϵ controls update magnitude, $[\cdot]_{\text{old}}$ indicates policy at last step, β is the influence of KL divergence \mathcal{D}_{KL} , whose detailed definition can be found in the section **Detailed equations** in Appendix.

Limitations of GRPO for small-size LLMs. Despite the success of GRPO in large language models (LLMs), small-size LLMs (SLMs) face significant challenges when confronted with complex problems requiring long chains of thought Zhang et al. [2025]. Due to their inherently limited capacity, SLMs struggle to generate high-quality, reward-worthy candidates for such tasks Li et al. [2025a], Zheng et al. [2025]. As shown in Figure 4a, where Qwen3-1.7B is implemented on a code task and it fails to generate correct answers for most queries. This limitation substantially reduces the probability of sampling high-reward candidates, resulting in advantage signals vanishing (Figure 5b), thereby constraining the potential performance gains achievable through GRPO in SLMs.

4 Methodology

To address the limitations of GRPO on SLMs, we propose incorporating guidance mechanisms into the thinking trajectories, thereby facilitating the sampling of high-quality candidates. We then conduct a comprehensive investigation into various design choices for guidance strategies. Finally, we introduce the G²RPO-A algorithm, which integrates our empirical observations and significantly reduces the need for extensive hyperparameter tuning.

4.1 Guided GRPO as a Solution

The Guided GRPO can be formulated as:

$$\begin{aligned} \mathcal{L}_{\text{guided}}(\theta) &= \mathbb{E}_{(q,a) \sim \mathcal{D}, \{g_i\}_{i=1}^G \sim \mathcal{G}, \{o_i\}_{i=1}^G \sim \pi_{\text{ref}}(\cdot | q, g_i)} \\ &\left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i| + |g_i|} \left(\sum_{t=1}^{|g_i|} \min(w_{g,i,t} \hat{A}_{i,t}, \text{clip}) \hat{A}_{i,t} + \sum_{t=1}^{|o_i|} \min(w_{o,i,t} \hat{A}_{i,t}, \text{clip}) \hat{A}_{i,t} - \beta \mathcal{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \right) \right], \end{aligned}$$

where $w_{g,i,t}$ and $w_{o,i,t}$ denote the token-level weighting coefficient of guidance g_i and the model outputs o_i , respectively. As shown in Figure 4b, this guidance enables SLMs to generate higher-reward candidates, potentially overcoming their inherent limitations.

Naive Guided GRPO fails to boost the final performance. Despite increasing expected rewards (Figure 4b), we found that

simply adding guidance to thinking trajectories of all candidates doesn't enhance final performance and suffers from low advantage.

As shown in Figure 5a and 5b, we train Qwen-3-1.7B-Base on a math dataset sourced from math 220k dataset Wang et al. [2024], and find that: (1) Guided GRPO's accuracy reward curve almost matches original GRPO. (2) Guided GRPO suffers from low advantage standard deviation, hindering the optimization of the models. As a result, further investigation is needed to leverage Guided GRPO's higher rewards while ensuring effective training, as the naive approach fails to utilize its potential benefits.

4.2 Optimizing Guided GRPO Design

In this section, we thoroughly examine optimal design choices for Guided GRPO, focusing on guidance ratio of GRPO candidate groups and adjusting guidance strength at different training stages. These investigations aim to maximize the effectiveness of the Guided GRPO and overcome the limitations observed in the naive implementation.

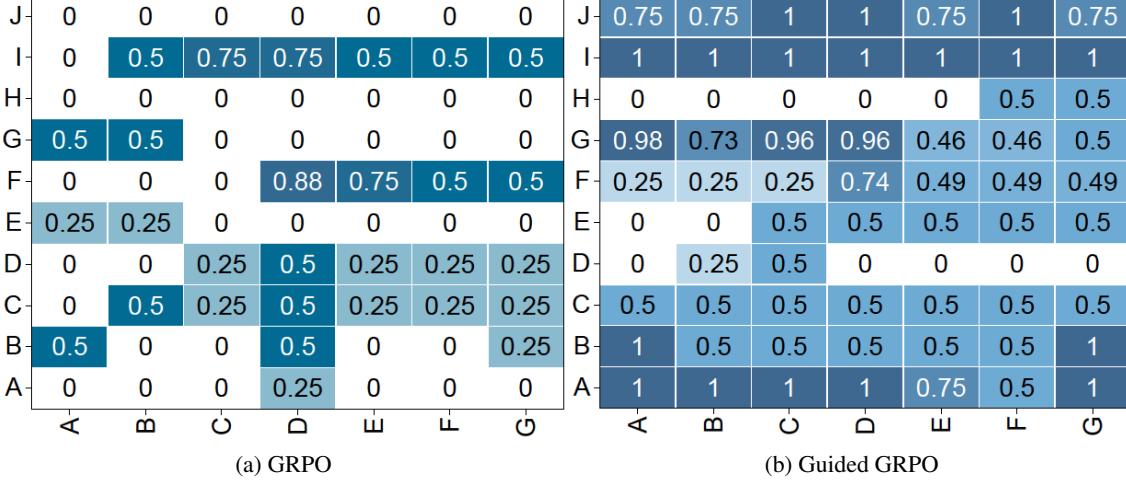


Figure 4: **Reward of Guided GRPO.** We fine-tuned Qwen3-1.7B on coding tasks, using 10 roll-outs and generating 280 candidates per batch. The candidates’ rewards form a 10×7 matrix. We then applied 2×2 average pooling, reducing it to a 10×7 matrix for clearer visualization. The results demonstrate that when configured with an optimal guidance ratio, G²RPO-A enables the model to sample candidates that yield a significantly denser reward signal.

MATH500	$\alpha = \frac{5}{6}$	$\alpha = \frac{3}{6}$	$\alpha = \frac{1}{6}$	$\alpha = 1$
$\ell = 50$	66.80	66.00	67.20	65.10
$\ell = 100$	65.20	63.00	66.20	64.70
$\ell = 200$	57.60	52.40	62.00	59.30
$\ell = 500$	57.80	62.00	68.20	55.80
$\ell = 0$		62.00		

Table 1: **Empirical study on guidance length ℓ and guidance ratio α .** We use the Qwen2.5-Math-7B as the backbone.

Inner-Group Varied Guidance Ratio. The insufficiency of naive guidance suggests that a more nuanced approach is required. We begin by investigating the impact of the guidance ratio α . In each GRPO group of size G , we steer only an α -fraction of the candidates. Let g_i denote the guidance for the i -th candidate (ordered arbitrarily), we have:

$$|g_i| = 0 \quad (i > \alpha G), \quad |g_i| = l \quad (i \leq \alpha G). \quad (3)$$

That is, the first αG candidates have guidance, while the remaining $(1 - \alpha)G$ candidates evolve freely. We conduct experiments on the Qwen2.5-Math-7B model Yang et al. [2024] with a roll-out number $n = 6$, training for one epoch on the s1k-1.1 dataset Muennighoff et al. [2025b]. We set $\alpha \in \{1/6, \dots, 1\}$ and $l \in \{50, 100, \dots, 500\}$ tokens, with all accuracies reported on the Math 500 benchmark. The results in Table 1 show that:

- *Partial inner-group guidance improves model performance.* In most settings, Guided GRPO with guidance provided to only a subset of candidates outperforms the vanilla GRPO, confirming the usefulness of the guidance mechanism.
- *For Qwen2.5-Math-7B on the Math500 benchmark, the lowest guidance ratio α combined with a longest guidance window ℓ yields the best results.* This suggests that Qwen2.5-Math-7B benefits from infrequent but heavyweight guidance.

In summary, selective guidance—directing only few candidates by a long guidance—strikes the best balance between exploration and control, thereby improving model performance. Moreover, the optimal guidance ratio varies with both the task domain and model capacity. As Table 8, 9 shows, smaller models and coding tasks benefit from stronger intervention, whereas larger models and math tasks achieve better results with lighter guidance.

Time Varied Guidance Length. Apart from the guidance ratio, Table 1 shows that performance also depends on the guidance length ℓ . To investigate this further, we evaluate guided GRPO by varying the guidance length during training

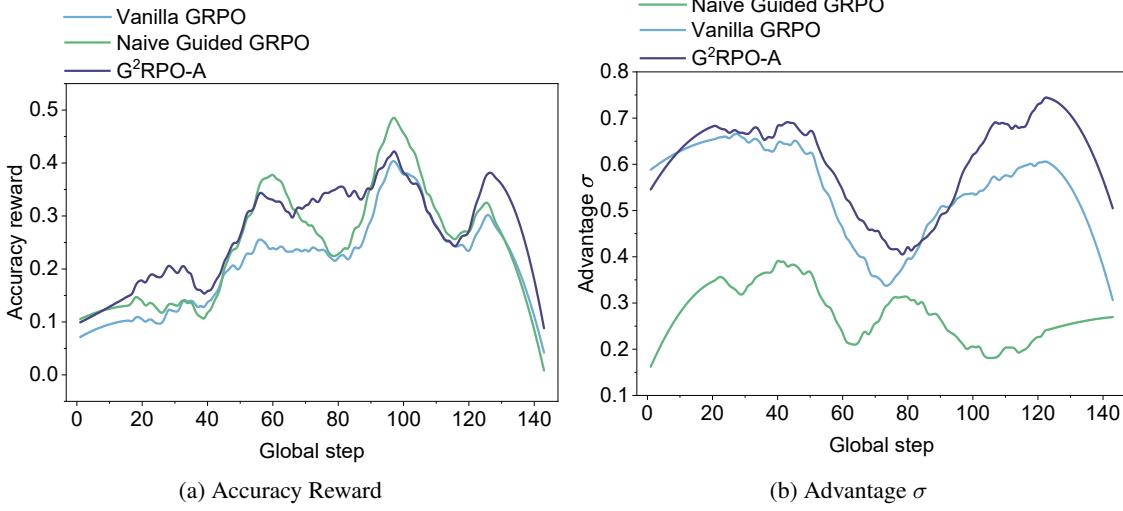


Figure 5: **Pitfalls of naive Guided GRPO.** We trained Qwen3-1.7B-Base on a curriculum-ordered subset of Math-220K Wang et al. [2024]: problems are presented from easy to hard. Because the curriculum continually increases task difficulty, the accuracy reward does not plateau at a high level—an expected outcome of the CL schedule. This training dynamic indicates that the advantage standard deviation is extremely low under the naive guidance condition, a situation that negatively impacts training efficiency for SLMs.

Guidance length	Decay Policies		
	Step	Linear	Concave
Guidance ratio			
$\ell = 50, \alpha = 0.8333$	63.80	58.40	57.60
$\ell = 100, \alpha = 0.1667$	60.60	62.00	66.20
$\ell = 200, \alpha = 0.8333$	66.60	54.20	58.40
$\ell = 200, \alpha = 0.1667$	61.20	64.20	59.60
$\ell = 500, \alpha = 0.1667$	59.60	69.80	62.40
$\ell = 0$	62.00		

Table 2: **Performance of Guided GRPO under different guidance-length adjustment policies.** We train Qwen2.5-Math-7B and evaluate it on the MATH 500 benchmark. For each guidance-length schedule, we report the results obtained with the guidance ratio that achieves the highest score in Table 1.

under three strategies. Those are:

$$\begin{aligned} \text{Concave decay: } \ell_t &= \ell_0 \left(1 - \frac{t}{T}\right)^\beta \\ \text{Linear decay: } \ell_t &= \ell_0 \left(1 - \frac{t}{T}\right) \\ \text{Stepwise decay: } \ell_t &= \ell_0 \gamma^{\lfloor t/s \rfloor}, \end{aligned} \tag{4}$$

where T is the total training steps, and ℓ_0 is the initial guidance length. The parameter $\beta \in (1, \infty]$ controls the concavity, and $\gamma \in (0, 1)$ sets the decay rate, and s specifies the decay interval.

We use the same experiment setting as in Table 1, and choose the guidance ratio that performs the best. The results are reported in Table 2. The results indicate that (1) model quality is highly sensitive to the chosen guidance length ℓ_t , and (2) no single schedule consistently outperforms the others. *This highlights the need for more effective methods of controlling guidance length.*

4.3 G²RPO-A: Sampling Difficulty Motivated Adaptive Guidance

In this section, we propose an adaptive algorithm that automatically selects the guidance strength ℓ at every optimization step. Our approach is inspired by recent work on data filtering and sampling [Bae et al., 2025, Xiong et al., 2025, Shi et al., 2025], which removes examples that yield uniformly low or uniformly high rewards. Such “uninformative”

Base Model	α	Benchmark	Base	GRPO	SFT	G²RPO-A
Qwen3-0.6B-Base	0.75	MATH500	40.18	54.26	50.53	51.77
		Minerva	11.43	9.57	10.40	12.29
		gpqa	25.49	24.51	25.49	30.39
Qwen3-1.7B-Base	0.25	MATH500	50.96	63.74	62.11	67.21
		Minerva	13.84	16.19	18.89	15.10
		gpqa	27.45	29.41	24.51	32.35
Qwen3-8B-Base	0.14	MATH500	71.32	79.49	80.29	82.08
		Minerva	33.24	37.51	36.60	36.42
		gpqa	43.17	44.13	42.85	49.72

Table 3: **Performance of G²RPO-A on Math Tasks.** We report accuracy (%) on various benchmarks. Models are trained for 5 epochs, and guidance ratios are selected based on the best settings obtained from Table 9.

Base Model	α	Benchmark	Base	GRPO	G²RPO-A
Qwen3-0.6B	0.75	MATH500	76.20	85.37	87.15
		Minerva	12.32	20.59	21.57
		gpqa	24.51	25.45	26.43
		AIME24	10.00	6.67	10.00
		AIME25	13.33	20.00	23.33
Qwen3-1.7B	0.25	MATH500	92.71	94.52	91.69
		Minerva	33.16	35.38	38.26
		gpqa	48.23	51.68	55.27
		AIME24	46.67	56.67	63.33
		AIME25	36.67	50.00	53.33

Table 4: **Performance of G²RPO-A on Math Tasks.** The experiment settings are the same with Table 3. However, we use extra benchmarks like AIME24 and AIME25 here due to the stronger model performances.

samples—being either too easy or too hard—contribute little to learning and can even destabilize training. The pseudo-code can be found in Appendix.

Guidance length adjustment. Our key idea is to control the difficulty of training samples by dynamically adjusting the guidance length, taking into account the ongoing training states. At each training step k , the guidance ℓ_{k+1} is determined by the following equation:

$$\ell_{k+1} = \ell_k \cdot \frac{\min(\mathcal{T}, k)r_k}{\sum_{\tau=1}^{\min(\mathcal{T}, k)} r_{k-\tau}}, \quad (5)$$

where r_k is the average reward of the k -th training step, \mathcal{T} is a hyperparameter that controls the number of history steps we considered, and we found that setting $\mathcal{T} = 2$ is already sufficient for noticeably improving Guided GRPO performance (Table 10, 11).

Equation 5 implies the following dynamics:

- When recent rewards rise, ℓ_k is reduced, making the next batch of examples harder.
- When recent rewards fall, ℓ_k is increased, making the next batch easier.

Thus, the training difficulty is automatically and continuously adjusted to match the model’s current competence.

Curriculum learning for further improvements. Equation 5 shows that the adaptive guidance-length controller updates ℓ by comparing the current reward with rewards from previous steps. When consecutive batches differ markedly in difficulty, these reward variations no longer reflect the model’s true learning progress, which in turn degrades G²RPO-A’s performance.

To eliminate this mismatch, we embed a curriculum-learning (CL) strategy Parashar et al. [2025], Shi et al. [2025], Zhou et al. [2025]. Concretely, we sort the samples by difficulty. Using math task as an example, we rank examples by source, yielding five ascending difficulty tiers: cn_contest, aops_forum, amc_aime, olympiads, and olympiads_ref. We also tested ADARFT Shi et al. [2025], which orders samples by success rate, but its buckets proved uninformative in

Base Model	Guidance Ratio	Benchmark	Base Perf.	GRPO	SFT	G^2RPO-A
Qwen3-0.6B	0.75	humaneval Live Code bench	32.32 17.07	38.89 22.22	40.33 13.58	44.96 23.14
Qwen3-1.7B	1	humaneval Live Code bench	46.08 34.31	67.65 53.14	63.34 56.33	75.93 51.96
Qwen3-8B	0.57	humaneval Live Code bench	64.36 60.58	81.48 77.12	77.42 63.82	80.33 79.71

Table 5: **Performance of G^2RPO-A on Code Tasks.** We report accuracy (%) on various benchmarks. Models are trained for 5 epochs, and guidance ratios are selected based on the best settings obtained from Table 8.

	Random		CL	
	GRPO	G^2RPO-A	GRPO	G^2RPO-A
Qwen3-1.7-Base				
MATH500	53.81	57.67	52.05	58.94
Minarva	12.41	15.12	14.98	16.69
gpqa	24.79	23.53	27.45	25.49
Qwen3-0.6B-Base				
Math 500	43.25	50.72	48.16	53.59
Minarva	11.04	11.21	9.66	10.08
gpqa	23.1	25.49	24.51	32.35

Table 6: **Comparison of training with random order and curriculum learning (CL) order** across different models and benchmarks.

Setting	Level 1	Level 2	Level 3	Level 4	Level 5
Remove	76.74	71.11	50.00	35.00	24.00
Replace	88.37	75.56	54.00	37.00	18.00
Original	86.05	77.77	60.00	43.00	28.00

Table 7: **Performance of GRPO with different sample-filtering methods.** We train Qwen3-1.7B model using G^2RPO-A , with $\alpha = 0.25$. In the REMOVE setting all hard samples are excluded from the original dataset, whereas in the REPLACE setting each hard sample is substituted with a sample of moderate difficulty.

our cases—most questions were either trivial or impossible (see Appendix Figure 6)—so it failed to separate difficulty levels effectively. Table 6 shows that both the performance of vanilla GRPO and G^2RPO-A boosted by CL.

Compare G^2RPO-A to sample-filtering methods. Earlier work argues that policy-gradient training benefits most from mid-level queries. Bae et al. [2025] keep only moderate-difficulty batches via an online filter, and Reinforce-Rej Xiong et al. [2025] discards both the easiest and hardest examples to preserve batch quality. Our experiments show that this exclusion is counter-productive: removing hard problems deprives the model of vital learning signals and lowers accuracy on challenging tasks. Table 7 confirms that either dropping hard items or substituting them with moderate ones reduces Level 4 and 5 test accuracy. G^2RPO-A avoids this pitfall by retaining tough examples and attaching adaptive guidance to them, thus exploiting the full difficulty spectrum without sacrificing performance.

5 Experiments

5.1 Experiment Settings

In this section, we outline the experiment settings we used, and more details about dataset filtering methods and evaluation on more models can be found in Appendix.

Datasets and models. We conduct experiments on math and code tasks. In detail,

Qwen3-1.7B	$\alpha = 0$	$\alpha = \frac{1}{4}$	$\alpha = \frac{2}{4}$	$\alpha = \frac{3}{4}$	$\alpha = 1$
humaneval	68.52	59.88	64.81	72.22	70.81
LCB	28.43	19.61	23.53	30.39	35.72
Qwen3-0.6B	$\alpha = 0$	$\alpha = \frac{1}{4}$	$\alpha = \frac{2}{4}$	$\alpha = \frac{3}{4}$	$\alpha = 1$
humaneval	41.98	32.10	27.72	38.89	49.38
LCB	12.75	11.76	9.80	18.63	12.75

Table 8: **Ablation studies on guidance ratio α for Code Tasks.** The group size is set to 12. The initial guidance length for G²RPO-A is set to 3072. The LCB indicates Live Code Bench.

	$\alpha = 0$	$\alpha = \frac{1}{4}$	$\alpha = \frac{2}{4}$	$\alpha = \frac{3}{4}$	$\alpha = 1$
Qwen3-1.7B-Base					
MATH500	52.05	58.71	53.09	55.53	45.95
Minerva	14.98	16.69	16.25	18.21	16.11
gpqa	27.45	25.49	30.39	25.49	22.55
Qwen3-0.6B-Base					
MATH500	48.16	49.59	50.94	53.50	38.42
Minerva	9.66	9.10	8.96	10.08	15.69
gpqa	24.51	19.61	31.37	32.35	25.49

Table 9: **Ablation studies on guidance ratio α for Math Tasks.** The group size is set to 12. The initial guidance length for G²RPO-A is set to 3072.

- **Mathematical reasoning tasks.** We construct a clean subset of the Open-R1 math-220k corpus Wang et al. [2024]. Problems are kept only if their solution trajectories are (i) complete, (ii) factually correct, and (iii) syntactically parseable.
- **Code generation.** For programming experiments we adopt the Verifiable-Coding-Problems-Python benchmark from Open-R1. For every problem we automatically generate a chain-of-thought with QWQ-32B-preview Team [2024]. These traces are later used as guidance by our proposed G²RPO-A training procedure.

We use Qwen3 series Yang et al. [2025a] for both tasks. Results of DeepSeek-Math-7B-Base Shao et al. [2024] for math and DeepSeek-Coder-6.7B-Base Guo et al. [2024] for code also included in Appendix. Unless specifically mentioned, CL is used for all experiments for fair comparison, and we also conducted ablation studies in Table 6.

Evaluation protocol. We assess our models mainly on three public mathematical-reasoning benchmarks—MATH500 Hendrycks et al. [2021], MINERVA-MATH Lewkowycz et al. [2022], and GPQA Rein et al. [2024]. For the mathematical training of Qwen3-1.7B and Qwen3-0.6B, AIME24 Li et al. [2024] and AIME25 benchmarks are also used. And for code tasks, we use two benchmarks: HUMANEVAL Chen et al. [2021] and LIVE CODE BENCH Jain et al. [2024]. Decoding hyper-parameters are fixed to: temperature = 0.6, top- p = 0.95, and top- k = 20. Unless otherwise noted, we generate with a batch size of 128 and permit a token budget between 1,024 and 25,000, based on each model’s context window.

Training details. Our G²RPO-A algorithm is implemented on top of the fully open-source OPEN-R1 framework Face [2025]. We use the following hyper-parameters: (i) number of roll-outs per sample set to 12 for 0.6B and 1.7B backbones, and 7 for 7B and 8B backbones; (ii) initial learning rate 1×10^{-6} , decayed with a cosine schedule and a warm-up ratio of 0.1; (iii) a training set of 1,000 problems for 5 epochs. Note that for ablation experiments, only 1 epoch is implemented in our training. (iv) All models are trained on 8 A100 GPUs.

5.2 Numerical Results

Superior performance of G²RPO-A. As reported in Table 3, 4, and 5, (i) our proposed G²RPO-A markedly surpasses vanilla GRPO on nearly every benchmark, and (ii) all RL-based methods outperform both the frozen base checkpoints and their SFT variants, mirroring trends previously observed in the literature.

GRPO	Fixed Guidance			RDP	G ² RPO-A
	3072	2048	1024		
Qwen3-1.7B-Base					
MATH500	52.05	51.28	60.52	46.78	51.02
Minerva	14.98	14.40	17.16	12.22	17.99
gpqa	27.45	25.00	24.51	23.53	22.13
					25.49
Qwen3-0.6B-Base					
MATH500	48.16	55.80	54.17	52.69	55.97
Minerva	9.66	13.27	15.26	11.78	14.32
gpqa	24.51	24.00	21.57	22.55	26.00
					32.35

Table 10: **Guidance-length ablation on Math Tasks.** Each run uses the optimal guidance ratio reported in Table 9. The initial guidance budget for G²RPO-A is fixed at 3,072 tokens. RDP refers to the rule-based decay policy.

GRPO	Fixed Guidance			RDP	G ² RPO-A
	3072	2048	1024		
Qwen3-1.7B					
humaneval	68.52	58.64	58.02	60.49	69.29
LCB	23.53	29.41	28.43	31.37	26.47
					70.81
Qwen3-0.6B					
humaneval	38.89	43.93	36.54	38.40	42.27
LCB	12.75	13.73	10.78	9.80	11.67
					12.75

Table 11: **Guidance-length ablation on Code Tasks.** Each run uses the optimal guidance ratio reported in Table 8. The initial guidance budget for G²RPO-A is fixed at 3,072 tokens. RDP refers to the rule-based decay policy.

Effect of the guidance ratio α . Table 8, 9 show that (1) larger models benefit from weaker guidance—e.g., Qwen3-1.7B peaks at $\alpha=0.25/0.5$ on Math, whereas the smaller Qwen3-0.6B prefers $\alpha=0.75$; (2) Code tasks consistently require a higher guidance ratio than Math.

Ablation on guidance-length schedules. Table 10, 11 contrast our adaptive scheme (G²RPO-A) with (i) fixed guidance and (ii) a rule-based decay policy (RDP). (1) G²RPO-A achieves the best score on almost every model–benchmark pair, confirming the benefit of on-the-fly adjustment. (2) For fixed guidance, the optimal value varies across both tasks and model sizes, with no clear global pattern, underscoring the need for an adaptive mechanism such as G²RPO-A.

6 Conclusion and Future Work

We introduce a method that injects ground-truth guidance into the thinking trajectories produced during GRPO roll-outs, thereby improving the performance of small-scale LLMs. After an extensive study of guidance configurations, we observe that the guidance ratio is significant in guidance mechanism and the optimal guidance length is context-dependent and, based on this, we develop G²RPO-A, an auto-tuned approach. Experiments on mathematical reasoning and code generation demonstrate that G²RPO-A consistently boosts accuracy. In future work, we plan to evaluate G²RPO-A across a broader range of tasks and model architectures, which we believe will further benefit the community.

References

- Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning oriented reinforcement learning, 2025. URL <https://arxiv.org/abs/2504.03380>.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas

- Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL <https://arxiv.org/abs/2107.03374>.
- Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *The Second Conference on Parsimony and Learning (Recent Spotlight Track)*, 2025. URL <https://openreview.net/forum?id=d3E3LWmTar>.
- 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. *CoRR*, 2025.
- Quy-Anh Dang and Chris Ngo. Reinforcement learning for reasoning in small llms: What works and what doesn't, 2025. URL <https://arxiv.org/abs/2503.16219>.
- Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL <https://github.com/huggingface/open-r1>.
- Xinyu Guan, Li Lyra Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rstar-math: Small LLMs can master math reasoning with self-evolved deep thinking. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=5zwF1GizFa>.
- Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Y Wu, YK Li, et al. Deepseek-coder: When the large language model meets programming-the rise of code intelligence. *CoRR*, 2024.
- 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.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL <https://openreview.net/forum?id=7Bywt2mQsCe>.
- Dong HUANG, Guangtao Zeng, Jianbo Dai, Meng Luo, Han Weng, Yuhao QING, Heming Cui, Zhijiang Guo, and Jie Zhang. Efficoder: Enhancing code generation in large language models through efficiency-aware fine-tuning. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=8bga0g1TlZ>.
- Qihan Huang, Weilong Dai, Jinlong Liu, Wanggui He, Hao Jiang, Mingli Song, Jingyuan Chen, Chang Yao, and Jie Song. Boosting mllm reasoning with text-debiased hint-grpo, 2025. URL <https://arxiv.org/abs/2503.23905>.
- 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. *CoRR*, 2024.
- Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *CoRR*, 2024.
- Ruipeng Jia, Yunyi Yang, Yongbo Gai, Kai Luo, Shihao Huang, Jianhe Lin, Xiaoxi Jiang, and Guanjun Jiang. Writing-zero: Bridge the gap between non-verifiable tasks and verifiable rewards, 2025. URL <https://arxiv.org/abs/2506.00103>.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.
- Jung Hyun Lee, June Yong Yang, Byeongho Heo, Dongyoon Han, and Kang Min Yoo. Token-supervised value models for enhancing mathematical reasoning capabilities of large language models. *CoRR*, 2024.
- Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh 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 Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL <https://openreview.net/forum?id=IFXTZERXdM7>.
- Chen Li, Nazhou Liu, and Kai Yang. Adaptive group policy optimization: Towards stable training and token-efficient reasoning. *arXiv preprint arXiv:2503.15952*, 2025a.
- Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions, 2024.

- Xuefeng Li, Haoyang Zou, and Pengfei Liu. Torl: Scaling tool-integrated rl. *arXiv preprint arXiv:2503.23383*, 2025b.
- Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. Cppo: Accelerating the training of group relative policy optimization-based reasoning models, 2025. URL <https://arxiv.org/abs/2503.22342>.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective, 2025. URL <https://arxiv.org/abs/2503.20783>.
- Zhenyan Lu, Xiang Li, Dongqi Cai, Rongjie Yi, Fangming Liu, Xiwen Zhang, Nicholas D Lane, and Mengwei Xu. Small language models: Survey, measurements, and insights. *CoRR*, 2024.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In *Workshop on Reasoning and Planning for Large Language Models*, 2025a. URL <https://openreview.net/forum?id=LdH0vrgAHm>.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025b. URL <https://arxiv.org/abs/2501.19393>.
- Vaskar Nath, Elaine Lau, Anisha Gunjal, Manasi Sharma, Nikhil Baharte, and Sean Hendryx. Adaptive guidance accelerates reinforcement learning of reasoning models, 2025. URL <https://arxiv.org/abs/2506.13923>.
- Chien Van Nguyen, Xuan Shen, Ryan Aponte, Yu Xia, Samyadeep Basu, Zhengmian Hu, Jian Chen, Mihir Parmar, Sasidhar Kunapuli, Joe Barrow, Junda Wu, Ashish Singh, Yu Wang, Jiuxiang Gu, Franck Dernoncourt, Nesreen K. Ahmed, Nedim Lipka, Ruiyi Zhang, Xiang Chen, Tong Yu, Sungchul Kim, Hanieh Deilamsalehy, Namyong Park, Mike Rimer, Zhehao Zhang, Huanrui Yang, Ryan A. Rossi, and Thien Huu Nguyen. A survey of small language models, 2024. URL <https://arxiv.org/abs/2410.20011>.
- Shubham Parashar, Shurui Gui, Xiner Li, Hongyi Ling, Sushil Vemuri, Blake Olson, Eric Li, Yu Zhang, James Caverlee, Dileep Kalathil, and Shuiwang Ji. Curriculum reinforcement learning from easy to hard tasks improves llm reasoning, 2025. URL <https://arxiv.org/abs/2506.06632>.
- Jinyoung Park, Jeehye Na, Jinyoung Kim, and Hyunwoo J. Kim. Deepvideo-r1: Video reinforcement fine-tuning via difficulty-aware regressive grp, 2025. URL <https://arxiv.org/abs/2506.07464>.
- 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 *First Conference on Language Modeling*, 2024.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL <https://arxiv.org/abs/1707.06347>.
- 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>.
- Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. Efficient reinforcement finetuning via adaptive curriculum learning, 2025. URL <https://arxiv.org/abs/2504.05520>.
- Débora Souza, Rohit Gheyi, Lucas Albuquerque, Gustavo Soares, and Márcio Ribeiro. Code generation with small language models: A deep evaluation on codeforces, 2025. URL <https://arxiv.org/abs/2504.07343>.
- Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown. *Hugging Face*, 2024.
- Jun Wang, Meng Fang, Ziyu Wan, Muning Wen, Jiachen Zhu, Anjie Liu, Ziqin Gong, Yan Song, Lei Chen, Lionel M. Ni, Linyi Yang, Ying Wen, and Weinan Zhang. Openr: An open source framework for advanced reasoning with large language models, 2024. URL <https://arxiv.org/abs/2410.09671>.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.
- Yuxiang Wei, Olivier Duchenne, Jade Copet, Quentin Carbonneaux, Lingming Zhang, Daniel Fried, Gabriel Synnaeve, Rishabh Singh, and Sida I Wang. Swe-rl: Advancing llm reasoning via reinforcement learning on open software evolution. *arXiv preprint arXiv:2502.18449*, 2025.
- Xumeng Wen, Zihan Liu, Shun Zheng, Zhijian Xu, Shengyu Ye, Zhirong Wu, Xiao Liang, Yang Wang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms, 2025. URL <https://arxiv.org/abs/2506.14245>.

- Jialong Wu, Shaofeng Yin, Ningya Feng, and Mingsheng Long. Rlvr-world: Training world models with reinforcement learning, 2025. URL <https://arxiv.org/abs/2505.13934>.
- Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, and Hanze Dong. A minimalist approach to llm reasoning: from rejection sampling to reinforce, 2025. URL <https://arxiv.org/abs/2504.11343>.
- Haoran Xu, Baolin Peng, Hany Awadalla, Dongdong Chen, Yen-Chun Chen, Mei Gao, Young Jin Kim, Yunsheng Li, Liliang Ren, Yelong Shen, Shuohang Wang, Weijian Xu, Jianfeng Gao, and Weizhu Chen. Phi-4-mini-reasoning: Exploring the limits of small reasoning language models in math, 2025. URL <https://arxiv.org/abs/2504.21233>.
- 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. *CoRR*, 2024.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengan Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.
- Zhicheng Yang, Zhijiang Guo, Yinya Huang, Xiaodan Liang, Yiwei Wang, and Jing Tang. Treerpo: Tree relative policy optimization, 2025b. URL <https://arxiv.org/abs/2506.05183>.
- Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning, 2025. URL <https://arxiv.org/abs/2502.03387>.
- Qiyi Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *CoRR*, 2025.
- Yang Yue, 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?, 2025. URL <https://arxiv.org/abs/2504.13837>.
- Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. R1-vl: Learning to reason with multimodal large language models via step-wise group relative policy optimization. *CoRR*, 2025.
- Haizhong Zheng, Yang Zhou, Brian R. Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for llm reasoning via selective rollouts, 2025. URL <https://arxiv.org/abs/2506.02177>.
- Yuhang Zhou, Jing Zhu, Shengyi Qian, Zhuskai Zhao, Xiyao Wang, Xiaoyu Liu, Ming Li, Paheng Xu, Wei Ai, and Furong Huang. Disco balances the scales: Adaptive domain- and difficulty-aware reinforcement learning on imbalanced data, 2025. URL <https://arxiv.org/abs/2505.15074>.
- Xiale Zhuang, Peixian Ma, Zhikai Jia, Shiwei Liu, and Zheng Cao. A technical study into 0.5b reasoning language models, 2025. URL <https://arxiv.org/abs/2506.13404>.