GRPO-LEAD: A Difficulty-Aware Reinforcement Learning Approach for Concise Mathematical Reasoning in Language Models

Anonymous ACL submission

Abstract

Group Relative Policy Optimization (GRPO), which is widely adopted by R1-like reasoning models, has advanced mathematical reasoning. Nevertheless, GRPO faces challenges in reward sparsity, verbosity, and inadequate focus on problem difficulty. We propose GRPO-LEAD, enhancing GRPO with: (1) length-regularized rewards to encourage conciseness while maintaining accuracy; (2) explicit penalties for incorrect solutions to improve model precision; and (3) difficulty-aware advantage reweighting for robust generalization on challenging problems. Comprehensive evaluations demonstrate that GRPO-LEAD significantly improves reasoning accuracy, conciseness, and efficiency. Our approach achieves state-of-the-art performance for 14B-scale models, underscoring the synergy of our methods with appropriate model scale and high-quality data. Our source code, generated dataset, and models are available after the acceptance of this paper.

1 Introduction

Recently, R1-like reasoning models have attracted significant attention due to their impressive performance in solving challenging mathematical reasoning tasks through extensive chains of thought(Luo et al., 2025b; Wen et al., 2025). According to the technical report introducing R1(Guo et al., 2025), reinforcement learning (RL) fine-tuning plays a pivotal role in enabling this reasoning capability. In particular, Group Relative Policy Optimization (GRPO)(Shao et al., 2024), a novel RL approach for language models, has emerged as a promising alternative to traditional methods such as PPO(Schulman et al., 2017) and DPO(Rafailov et al., 2023), primarily due to its efficiency and intrinsic compatibility with language model training. . Researchers across various domains have successfully employed GRPO (Li et al., 2025; Liu et al., 2025a; Luo et al., 2025a; Dai et al., 2025), achieving impressive outcomes.

Despite its strengths, existing GRPO implementations encounter significant limitations.A primary issue is reward sparsity stemming from binary, rule-based accuracy metrics; when responses within problem groups exhibit uniform correctness or incorrectness, the resulting uniform reward signals offer minimal differentiation, weakening learning gradients and hampering convergence. Moreover, such uniform signals inadequately promote concise reasoning, leading to unnecessarily verbose outputs and inefficiencies during training and inference. Additionally, the current reward formulation lacks explicit penalties for incorrect answers(Hu et al., 2025a; Luo et al., 2025b; Chu et al., 2025),, inadvertently encouraging models to guess rather than engage in rigorous reasoning, thereby compromising precision. Furthermore, rewards are applied uniformly across problems regardless of their intrinsic difficulty, causing models to excessively optimize simpler tasks while neglecting more challenging problems that require deeper reasoning.

Furthermore, computational efficiency also emerges as a critical practical concern, as reinforcement learning fine-tuning typically demands substantial resources, limiting accessibility, experimentation speed, and scalability, especially in low-resource environments. The current GRPO formulation is insufficient for encouraging concise and precise reasoning. Consequently, reducing computational requirements during both training and inference is essential for enabling broader applicability and effective real-world deployment.

Motivated by these limitations, this work introduces GRPO-LEAD, a suite of targeted modifications explicitly designed to enhance GRPO's effectiveness for mathematical reasoning tasks. The overall framework is illustrated in figure 1. Our key contributions include:

[*] Equal contribution.

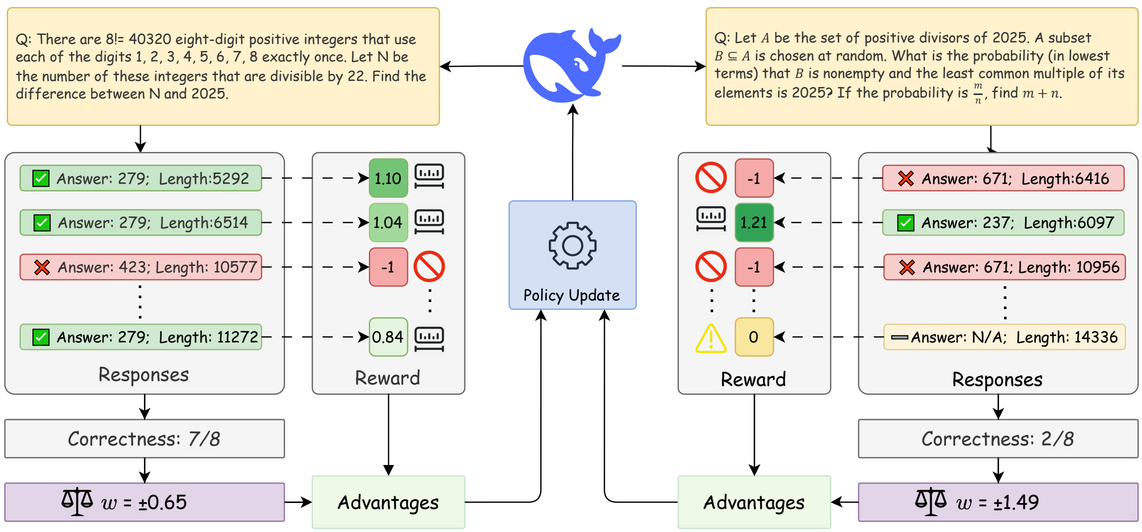


Figure 1: The GRPO-LEAD framework assigns length-regularized positive rewards to correct answers and explicit penalties to incorrect ones. A difficulty-based weight w used for advantage reweighting is determined from the empirical correctness of responses for each question. This weight then scales the advantages derived from each question, prioritizing harder questions over easier ones during the policy update to foster robust reasoning.

- We introduce a length-regularized reward with an explicit penalty for incorrect solutions to encourage solution conciseness while maintaining accuracy.
- We apply difficulty-aware advantage reweighting to focus learning on more challenging problems, fostering robust generalization.
- Our comprehensive evaluations demonstrate GRPO-LEAD significantly improves reasoning accuracy and conciseness, achieving state-of-the-art performance in mathematical reasoning for 14B-scale models.

2 Related Work

2.1 Group Relative Policy Optimization

Group Relative Policy Optimization (GRPO) is a recently proposed algorithm designed specifically for fine-tuning language models with group-level normalization of rewards (Guo et al., 2025). GRPO modifies the standard policy gradient objective by introducing relative advantages within sets of responses corresponding to the same query, stabilizing updates and promoting consistent learning signals. Formally, GRPO defines the objective as:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\Big[\min\big(r_{i,t}(\theta)\hat{A}_{i,t}, \tag{1}$$

$$\text{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_{i,t}\big)\Big]$$

where the importance sampling ratio is given by

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}. \tag{2}$$

Here, G denotes the number of groups (e.g., different queries), $\hat{A}_{i,t}$ is the normalized advantage within group i, and ϵ defines the clipping range for conservative updates.

2.2 Length Reward

A prevalent issue in reinforcement learning-based fine-tuning of language models is the uncontrolled increase in response length driven by reward signals, commonly known as *reward hacking*(Everitt et al., 2017; Gao et al., 2023; Weng, 2024). This phenomenon leads to unnecessarily verbose responses, which, although technically correct, often lack conciseness and hinder interpretability. Furthermore, such verbosity fails to reflect efficient reasoning, limiting model utility in practical scenarios. Existing efforts to mitigate this problem typically involve incentivizing shorter answers to encourage more succinct reasoning processes. For example, Kimi proposed an individual min-max normalized length reward based on the lengths of generated responses (Team et al., 2025). Yeo et al. introduced a cosine length reward function with fixed maximum and minimum thresholds to manage response lengths (Yeo et al., 2025). Aggarwal et al. utilized a target "golden length" to directly reward or penalize responses based on their deviation from an ideal length (Aggarwal and Welleck, 2025).

However, these existing methods depend heavily on static or predefined length heuristics, limiting their effectiveness across diverse questions of varying complexity. In contrast, our proposed length-dependent accuracy reward addresses these limitations by dynamically calibrating rewards according to each group's relative response length and rollout accuracy, promoting concise yet difficulty-aware reasoning processes.

3 Method

To systematically address the limitations identified in existing implementations of Group Relative Policy Optimization (GRPO), we propose a suite of novel modifications collectively termed **GRPO-LEAD** (**GRPO** with **L**ength-dependent rewards, **E**xplicit penalties, and **A**dvantage reweighting for **D**ifficulty). Our proposed method enhances the original GRPO framework by introducing three core innovations: 1) a length-dependent accuracy reward to foster concise solutions, 2) an explicit penalty mechanism to mitigate low precision rate caused by length reward, and 3) a difficulty-aware advantage reweighting strategy that amplifies learning signals for challenging problems. Additionally, we examine how base model scale and supervised fine-tuning (SFT) impact the effectiveness of reinforcement learning (RL) fine-tuning.

3.1 Length-Dependent Accuracy Reward

The core idea is to reward correct completions not uniformly but in proportion to their relative conciseness. Given a question q and a set of model-generated responses $\{o_i\}$, we first isolate the subset of correct responses and compute the mean μ and standard deviation σ of their token lengths. For a correct response o with length $|o|$, we define its standardized length deviation as:

$$z = \frac{|o| - \mu}{\sigma + \epsilon}, \tag{3}$$

where $\epsilon > 0$ is a small constant added for numerical stability. The final reward is modulated using an exponential decay function:

$$R_{\text{accuracy}}(o|q) = \begin{cases} \exp(-\alpha z), & \text{if } o \text{ is correct,} \\ 0, & \text{if } o \text{ is incorrect.} \end{cases} \tag{4}$$

where $\alpha > 0$ is a tunable hyperparameter controlling the strength of length penalization.

This formulation ensures that overly long correct responses are systematically penalized, while relatively concise ones are amplified. Unlike static or absolute length constraints, our approach leverages standardized deviation, allowing for dynamic adaptation to the distributional properties of each question.

3.2 Explicit Penalty for Incorrect Answers to Enhance True Accuracy

Existing methods often prioritize maximizing *pass@1*—the success rate on the first attempt—typically within restricted response lengths. However, this focus can inadvertently degrade overall model accuracy. The fundamental issue appears to stem from the use of a binary accuracy reward, rather than length-based regularization: under pressure to generate responses within a limited length, a model is encouraged to provide an answer, even if it's a guess, rather than no answer at all. Such guesses can achieve a non-zero reward and inflate *pass@1*, but they do so at the cost of overall precision by rewarding less rigorous reasoning.

To counteract this tendency and foster a more robust distinction between correct and incorrect outputs, we introduce a revised reward structure that explicitly penalizes incorrect responses. This new reward function is defined as:

$$R_{\text{accuracy}}(o \mid q) = \begin{cases} \exp(-\alpha z), & \text{if } o \text{ is correct,} \\ -1, & \text{if } o \text{ is incorrect,} \end{cases} \tag{5}$$

where o is the output, q is the question, z represents the standardized length deviation of a correct response, and $\alpha > 0$ is a hyperparameter controlling the strength of the length penalization for correct answers, consistent with prior definitions.

The expected reward for a response, given its probability of correctness $P(\text{correct})$, under this formulation is:

$$\mathbb{E}[R_{\text{accuracy}}(o \mid q)] = P(\text{correct}) \cdot \exp(-\alpha z) - (1 - P(\text{correct})) \tag{4}$$

To intuitively grasp the impact of this reward function, let us consider a simplified scenario where the length penalty for correct answers is negligible (i.e., $\exp(-\alpha z) \approx 1$). In practice, the average reward for correct answers often normalizes close to this value. Under this assumption, the expected reward simplifies to:

$$\mathbb{E}[R] \approx 2P(\text{correct}) - 1 \tag{5}$$

This approximation reveals a crucial characteristic: the expected reward becomes positive only when $P(\text{correct}) > 0.5$. This threshold acts as a principled deterrent against speculative guessing, compelling the model to internalize a more stringent decision boundary for correctness. Our empirical results confirm that this approach significantly improves both *pass@1* and overall precision, encouraging the model to favor accuracy over mere completion.

3.3 Advantage Reweighting for Difficulty-Aware Training

While length reward and advantage reweighting can enhance precision and mitigate verbosity, uniformly applying rewards across all questions, irrespective of their intrinsic difficulty, may implicitly bias the model. It might learn to excessively optimize performance on simpler tasks-—where correct and concise responses are more readily achieved-—while neglecting more complex questions that demand deeper reasoning. Consequently, the performance on challenging problems can degrade.

Therefore, we introduce a difficulty-aware advantage reweighting strategy, which dynamically adjust the magnitude of policy updates based on an estimate of problem difficulty. The intuition is to amplify learning signals for harder tasks, re-anchoring the model towards harder tasks.

Formally, we first quantify problem difficulty. For a given question q and its associated set of sampled responses $\{o_i\}$, we define the group's empirical correctness ratio as:

$$\rho_q = \frac{\text{number of correct responses for } q}{\text{total number of responses for } q}. \tag{6}$$

This ratio, ρ_q, serves as an inverse proxy for problem difficulty: a lower ρ_q suggests a harder question.

Next, we introduce a logistic reweighting factor dependent on this ratio to modulate the advantage estimates during the RL training step. The logistic function is defined as:

$$w(\rho_q) = A + \frac{B - A}{1 + \exp\left[k(\rho_q - \rho_0)\right]}, \tag{7}$$

where hyperparameters A, B, ρ_0, k allow precise control over the sensitivity of weighting to problem difficulty.

To apply this reweighting, we first consider the normalized advantage estimate for a response o_i to question q:

$$\tilde{A}_i = \frac{R(o_i|q) - \mu_q}{\sigma_q + \epsilon}, \tag{8}$$

where μ_q and σ_q are the mean and standard deviation of rewards $R(o_i|q)$ for responses to question q, and ϵ is a small constant for numerical stability. We then define the difficulty-aware advantage, A'_i, as:

$$A'_i = \tilde{A}_i \cdot \begin{cases} w(\rho_q), & \text{if } \tilde{A}_i > 0 \\ w(1 - \rho_q), & \text{if } \tilde{A}_i \leq 0 \end{cases} \tag{9}$$

This formulation ensures that for difficult problems (low ρ_q), correct responses (which are rare and thus highly valuable) receive substantially larger updates due to the increased weighting $w(\rho_q)$. Conversely, incorrect responses on easier problems (high ρ_q) are penalized more strongly, sharpening the decision boundary for problems where high performance should be expected.

3.4 Impact of Data Quality on Reinforcement Learning Effectiveness

To further enhance model capabilities, we first performed supervised fine-tuning (SFT) on a specialized dataset of 13k math reasoning problems sourced from DeepScaler(Luo et al., 2025b) (including historical AMC, AIME, and OmniMath problems) with solutions generated by QwQ32B(Team, 2025). Although this SFT model initially showed signs of overfitting, subsequent application of our proposed RL strategies rapidly mitigated these issues. This SFT+RL approach yielded faster convergence and significantly improved pass@1 accuracy and overall precision compared to applying RL directly to the original base model.

Our findings also highlight the critical role of data quality and curriculum strategies in RL. We established a robust initial policy by applying RL to a subset of challenging problems from the DeepScaler dataset. This policy was then further refined using a curriculum composed of the most challenging problems identified from this first RL stage and supplemented by high-difficulty examples from the Light-R1 dataset(Wen et al., 2025). This two-stage curriculum markedly enhanced the model's ability to continuously improve on complex tasks.

Finally, we addressed a persistent formatting issue of repetitive n-gram patterns, likely stemming from an absence of clear end-of-sequence (EOS) signals during SFT. By temporarily removing length-dependent rewards and introducing an explicit negative reward (-1.5) for such repeated n-grams, we achieved further improvements in precision and pass@1 metrics. This intervention demonstrates the effectiveness of targeted reward modifications for mitigating specific output anomalies.

In summary, our experiments affirm that initial model capacity, curated data curricula for RL, and targeted reward engineering are pivotal for optimizing fine-tuning outcomes. These elements collectively inform a systematic approach for enhancing language models' ability to produce concise, accurate, and well-structured responses across tasks of varying complexity.

4 Experimental Setup

We evaluate GRPO-LEAD, integrating length-dependent accuracy rewards, explicit penalties for incorrect solutions, and difficulty-aware advantage reweighting, on DEEPSEEK-R1 DISTILLED variants (Guo et al., 2025; Yang et al., 2024). Our experiments cover two model scales, 7B and 14B parameters. All GRPO training is conducted using the VERL framework.(Sheng et al., 2024).

4.1 Datasets and Filtering

Our primary training data is sourced from the DEEPSCALER dataset (Luo et al., 2025b). We filter out problems with difficulty ratings below 2.5, resulting in approximately 9,000 questions for fine-tuning.

For stages 2 of our 14B model experiments, we further refine the dataset by selecting problems where the model's stage-1 rollout accuracy is no greater than 75%, yielding around 2,283 questions. Additionally, we incorporate challenging problems with numeric answers from the stage-2 dataset of Light-R1 (Wen et al., 2025), resulting in 3,524 question in total.

4.2 Hyperparameters

We train with a learning rate of 1×10^{-6}, batch size 32, and group size 8–generating 8 rollouts per question for GRPO reward computation. The KL penalty term is removed, as it was found to suppress exploration in our experiments, which is also suggested in similar works(Liu et al., 2025b; Hu et al., 2025b).

For the length-dependent accuracy reward, we set $\alpha = 0.05$, providing a moderate decay that encourages conciseness without penalizing slight verbosity. For difficulty-aware advantage reweighting, we use $A = 0.4$, $B = 1.5$, $\rho_0 = 0.75$, and $k = 10$. This configuration ensures reweighting is minimal on easy problems but sharply increases near the 75% correctness threshold. The steep slope ($k = 10$) enables strong emphasis on high-difficulty examples, guiding the model to allocate learning more effectively.

4.3 Model Variants and Fine-Tuning Stages

7B Model Experiments Starting from the DeepSeek-R1 Distilled 7B Qwen-Math checkpoint, we first apply standard GRPO on the 9k questions, producing a baseline. Then, we train 3 more models from the DeepSeek-R1 Distilled 7B Qwen-Math checkpoint, adding one more of the following components subsequently: (i) Length Reward only, (ii) Length Reward + Advantage Reweighting, (iii) Length Reward + Advantage Reweighting + Explicit Penalty. We train for approximately 200 steps and select the top-performing checkpoints based on validation results. At test time, we limit the generation length to 8k for all 7B models, matching the training length limit.

14B Model Experiments We extend the above procedure to the DeepSeek-R1 Distilled 14B Qwen checkpoint across multiple stages. In **Stage 1**, we train for 100 steps using all GRPO-LEAD components on the filtered 9k-question dataset. To enhance the model's base capability, we first fine-tune the model on a curated set of 13k math problems with supervised fine-tuning (SFT), then conduct the RL phase. This SFT stage significantly improves the model's reasoning quality, even though it tends to increase the output length and caused some format error.

The SFT data consists of all problems in the DEEPSCALER dataset with difficulty greater than 1. To construct high-quality reasoning traces for SFT, we use the QWQ-32B model(Team, 2025) to generate step-by-step solutions.

After observing that some questions remain low correctness, we further fine-tune for **Stage 2** to focus on those underperformed problems. We also address the repetitive output patterns by removing the length penalty and introducing a negative reward (-1.5) for repeated n-grams. We continue training for 240 more steps (100 steps with initial settings and 140 more steps with repetition penalty), yielding the final model checkpoint. At test time, we limit the generation length to 14k for all 14B models, in accordance with our training settings and also to better compare the models' performance in a low-budget scenario.

4.4 Baselines and Evaluation Protocol

We compare our models with both DEEPSEEK-R1 DISTILLED-14B-QWEN(Guo et al., 2025) (the distilled Qwen model without GRPO-LEAD) and LIGHT-R1-14B-DS (Wen et al., 2025), which has the same base model as ours and was first finetuned with 3k hard math problems with SFT, and then fine-tuned with a cosine-based length reward (Yeo et al., 2025) on their selected math problems for three epochs using GRPO.

We primarily report three metrics: **(1) Cons@32**, accuracy through majority voting for 32 samplings; **(2) Pass@1**, the probability that the top-1 sample is correct under a chosen decoding strategy; **(3) Average Length** (Len_{avg}), measuring verbosity. Unless otherwise specified, we decode with temperature 0.6 and sample 32 solutions per question, then compute Cons@32 and Pass@1 over these samples.

5 Results

In this sction, we present a comprehensive evaluation of the proposed GRPO-LEAD framework on two mathematical benchmarks: AIME24 and AIME25. Our analysis is structured as follows: we first examine training dynamics to illustrate how GRPO-LEAD accelerates convergence; next, we perform an ablation study to assess the incremental benefits of each component; and finally, we compare against state-of-the-art baselines for 14B-scale language models.

5.1 Training Dynamics

Figure 2 plots the evolution of Pass@1 on a validation split over training steps for three configurations of the 7B model: (i) baseline GRPO, (ii) GRPO with length reward, and (iii) GRPO with both length reward and advantage reweighting. We observe two clear trends. First, adding a length-dependent reward not only yields higher Pass@1 but also accelerates early-stage convergence, suggesting that penalizing overly verbose correct solutions provides a more informative learning signal.

[*]The validation consists of 27 challenging problems from AIMO2 (Frieder et al., 2024), CMU-MATH-AIMO (Sun, 2024), and AIME24.

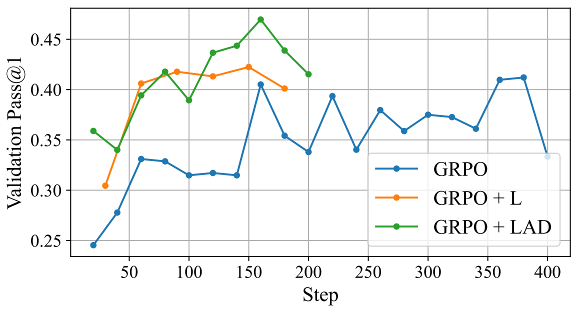


Figure 2: Validation[*] Pass@1 over training steps for three configurations: GRPO, GRPO+L, and GRPO+LAD. Shown by the faster convergence, Length Reward and Advantage Reweighting provides richer reward signal than the original setup.

Second, incorporating advantage reweighting (to amplify updates on harder questions) further steepens the trajectory, indicating that reweighting advantage estimates according to problem difficulty helps the model refine reasoning on challenging prompts more efficiently.

Overall, these dynamics confirm that GRPO-LEAD components—particularly the length reward—bolster training stability and speed. By comparison, the baseline GRPO model learns more slowly and lags behind in Pass@1 across the entire training horizon.

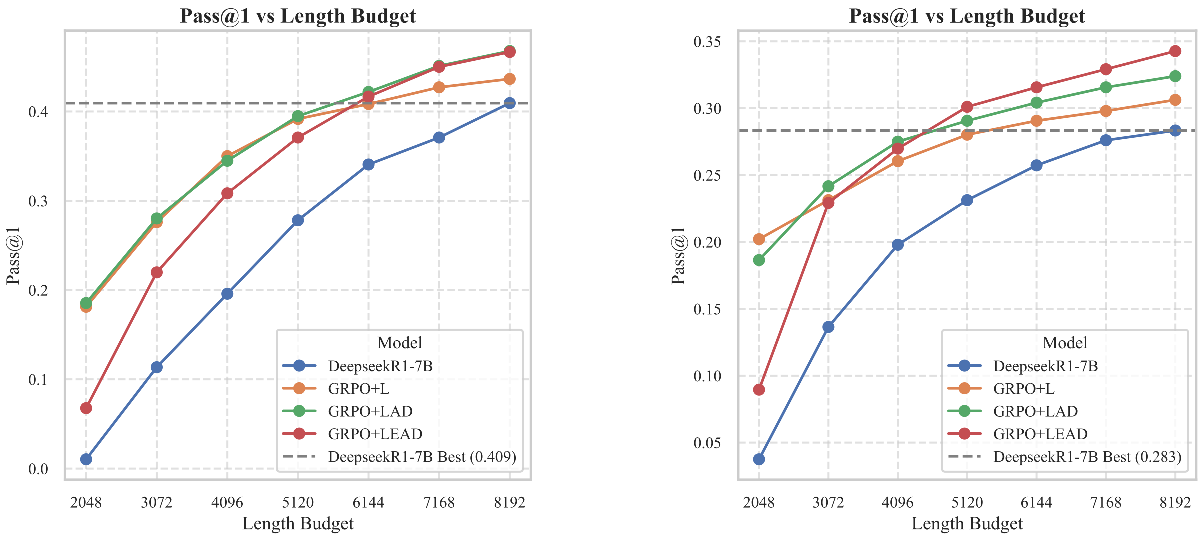
5.2 Ablation Analysis

We next quantify the contribution of each GRPO-LEAD component through a step-by-step ablation on the 7B model. Table 1 summarizes results on AIME24 and AIME25.

Length Reward Brings Conciseness to Reasoning We first incorporate the length-dependent accuracy reward into GRPO. Compared to Deepseek-7B, length reward slightly improves Pass@1 on both AIME24 ($0.431 \rightarrow 0.438$) and AIME25 ($0.292 \rightarrow 0.308$), with an additional improvement of Cons32 by 14.1% on AIME25. Notably, these improvements are accompanied by a substantial reduction of 1,715 tokens (24.5%) and 1,903 tokens (26.8%) in the average response length on the two datasets, respectively. Figure 3 further demonstrates that length reward largely enhances performance in low-budget settings over the base model, matching its peak performance with only 5/8 of the token budget on the more difficult AIME25. These results demonstrate that length reward, by penalizing correct but overly verbose solutions, can effectively reduce unnecessary text without com-

Ablation Setting	AIME24			AIME25		
	Cons@32	**Pass@1**	**Len$_{\text{avg}}$**	**Cons@32**	**Pass@1**	**Len$_{\text{avg}}$**
Deepseek-7B	0.767	0.431	6,990	0.467	0.292	7,113
GRPO + len. reward	0.767	0.438	**5,275**	0.533	0.308	**5,210**
+ adv. reweighting	0.767	0.458	5,323	**0.567**	0.325	5,437
+ explicit penalty	**0.800**	**0.470**	6,104	**0.567**	**0.345**	6,308

Table 1: Ablation results on AIME24 and AIME25. We report **Cons@32** (the fraction of problems for which at least one correct solution is found among 32 samples), **Pass@1**, and the average token length (**Len**$_{\text{avg}}$). The best value in each column is in boldface, the second best is underlined.



(a) AIME24 (b) AIME25

Figure 3: Performance against inference budget for training done with different ablations of LEAD. GRPO with length reward (GRPO+L) largely enhances the performance at low budget settings compared to before training (DeepseekR1-7B).

promising overall performance.

Advantage Reweighting Encourages Model to Solve more Difficult Problems Further incorporating difficulty-aware advantage reweighting (GRPO+LAD) refines performance. On AIME24, Pass@1 increases from the GRPO+L stage (0.438 $\rightarrow$ 0.458), while Cons@32 remains 0.767. For AIME25, both Pass@1 (0.308 $\rightarrow$ 0.325) and Cons@32 (0.533 $\rightarrow$ 0.567) see improvements. As Figure 3 shows, GRPO+LAD demonstrates gains over GRPO+L in almost all budget regimes on AIME25 and for budgets exceeding 5k tokens on AIME24. These results indicate that advantage reweighting, by prioritizing challenging problems, strengthens reasoning robustness and mitigates over-reliance on simpler examples, thus validating its role in driving more reliable generalization.

Explicit Penalty for Incorrect Answers Regularizes Thinking Finally, introducing an explicit penalty for incorrect solutions (GRPO+LEAD) yields the highest Pass@1 scores. On AIME24, Pass@1 improves from the GRPO+LAD stage (0.458 $\rightarrow$ 0.470) and Cons@32 climbs (0.767 $\rightarrow$ 0.800). On AIME25, Pass@1 also increases (0.325 $\rightarrow$ 0.345), as detailed in Table 1. Notably, these gains involve a modest increase in average solution length on AIME24 (from approximately 5,300 to 6,104 tokens). Figure 3 illustrates this trade-off, showing a performance sacrifice in low-budget regimes, though GRPO+LEAD still outperforms GRPO+LAD with budgets higher than 5k tokens on AIME25. These results suggest that the explicit penalty serves as a regularizer for the model to be more conservative about its reasoning. Such regularization boosts performance while requiring a slightly longer thinking process, which nevertheless remains shorter than the Deepseek-7B

Model Name	AIME24			AIME25		
	Cons@32	**Pass@1**	**Len**$_{\text{avg}}$	**Cons@32**	**Pass@1**	**Len**$_{\text{avg}}$
DeepSeek-14B	0.800	0.614	9,182	0.633	0.429	10,046
Light-R1-14B-DS	<u>0.833</u>	<u>0.641</u>	9,571	**0.767**	0.505	10,194
LEAD-stage1	<u>0.833</u>	0.629	<u>8,790</u>	**0.767**	<u>0.523</u>	<u>9,371</u>
LEAD-stage2	**0.867**	**0.650**	**8,267**	**0.767**	**0.539**	**8,668**

Table 2: Comparison of model performance on AIME24 and AIME25, showing **Cons@32**, **Pass@1**, and average token length (**Len**$_{\text{avg}}$). The best value in each column is in boldface, the second best is underlined.

baseline.

Overall, these ablation results confirm that all three enhancements—length-dependent accuracy, difficulty-aware advantage reweighting, and explicit penalties—collectively reduce verbosity, strengthen mathematical skills on harder questions, and elevate precision in final predictions.

5.3 Comparison with Baselines

We next evaluate GRPO-LEAD at the 14B scale and compare it against two strong baselines under a 14k-token generation budget: **DeepSeek-14B** and the state-of-the-art **Light-R1-14B-DS**. Table 2 presents results on AIME24 and AIME25, including both our intermediate model (*LEAD-stage1*) and our final model (*LEAD-stage2*).

AIME24 Performance LEAD-stage1 achieves a Cons@32 of 0.833, matching Light-R1-14B-DS and exceeding DeepSeek-14B by 4.1%. Its Pass@1 outperforms DeepSeek-14B by 2.4% and closely approaches Light-R1-14B-DS. Crucially, LEAD-stage1 produces more concise responses than both baselines, with more than 800 tokens less on average. Building on these gains, LEAD-stage2 pushes performance further, delivering the highest Cons@32 (4% above Light-R1-14B-DS) and the best Pass@1, while reducing average solution length to 8,267 tokens.

AIME25 Performance LEAD-stage1 yields a Cons@32 of 0.767, matching Light-R1-14B-DS and exceeding DeepSeek-14B by 21.2%. Its Pass@1 (0.523) outperforms DeepSeek-14B by 21.9% and Light-R1-14B-DS by 3.6%. Crucially, LEAD-stage1 produces more concise responses than both baselines, with its solutions averaging 9,371 tokens. Building on these gains, LEAD-stage2 pushes performance further, delivering the highest Cons@32 (matching Light-R1-14B-DS at 0.767) and the best Pass@1 (0.539), while reducing average solution length to 8,668 tokens.

Overall, both LEAD-stage1 and LEAD-stage2 deliver substantial improvements over DeepSeek-14B and Light-R1-14B-DS, simultaneously boosting correctness and conciseness under a constrained (14k-token) budget. Remarkably, training LEAD-stage1 for just 100 steps—requiring only about 24 hours on eight H20 GPUs—already matches Light-R1-14B-DS on Cons@32 and outperforms it on AIME25 Pass@1 while producing shorter solutions, underscoring the practical efficiency of GRPO-LEAD for large-scale math problem-solving.

6 Conclusion

We introduced GRPO-LEAD, a reinforcement learning framework designed for mathematical reasoning tasks. By extending Group Relative Policy Optimization with three major components—(1) a length-dependent accuracy reward to discourage overly verbose solutions, (2) an explicit negative penalty that clarifies the boundary between correct and incorrect answers, and (3) a difficulty-aware advantage reweighting scheme to prioritize tougher problems—GRPO-LEAD addresses key challenges in structured problem-solving.

Empirical evaluations on two AIME benchmarks show that GRPO-LEAD not only speeds up convergence but also strengthens the model's reasoning capability while keeping solution paths concise. Our 14B-scale experiments further confirm that GRPO-LEAD achieves state-of-the-art performance by balancing output brevity with high problem-solving accuracy. Although open questions remain—particularly in managing partial correctness and extending these techniques to broader domains—our findings suggest that reward shaping and difficulty modeling are pivotal in developing more robust and aligned language models for complex mathematical reasoning.

7 Limitations

Although our techniques for encouraging concise solutions and difficulty-balanced learning may transfer to other domains, the gains reported here are specific to mathematical reasoning tasks. Further studies are needed to evaluate the effectiveness of GRPO-LEAD on broader question-answering or logical reasoning domains, where correctness signals and domain structures can differ substantially.

Additionally, we only have access to a limited amount of compute, which prevents us from conducting more comprehensive experiments. For instance, we currently cannot provide the validation curve for the 7B model in the ablation study that adds an explicit penalty. This is due to an error in the validation code after upgrading to the newest VERL version, and we currently don't have the compute to reproduce it. The comparison with original GRPO model is also lacked except the curve shown in figure 2 since the checkpoint is on the server on the rented server, which was automatically released at the point we write the paper. We also couldn't formally perform a hyperparameter search to showcase the rationale behind choosing the hyperparameters for our designed modifications.

References

Pranjal Aggarwal and Sean Welleck. 2025. L1: Controlling how long a reasoning model thinks with reinforcement learning. arXiv preprint arXiv:2503.04697.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. 2025. Gpg: A simple and strong reinforcement learning baseline for model reasoning. arXiv preprint arXiv:2504.02546.

Muzhi Dai, Chenxu Yang, and Qingyi Si. 2025. S-grpo: Early exit via reinforcement learning in reasoning models. arXiv preprint arXiv:2505.07686.

Tom Everitt, Victoria Krakovna, Laurent Orseau, Marcus Hutter, and Shane Legg. 2017. Reinforcement learning with a corrupted reward channel. arXiv preprint arXiv:1705.08417.

Simon Frieder, Sam Bealing, Arsenii Nikolaiev, Geoff C. Smith, Kevin Buzzard, Timothy Gowers, Peter J. Liu, Po-Shen Loh, Lester Mackey, Leonardo de Moura, Dan Roberts, D. Sculley, Terence Tao, David Balduzzi, Simon Coyle, Alex Gerko, Ryan Holbrook, Addison Howard, and XTX Markets. 2024. Ai mathematical olympiad - progress prize 2. https://kaggle.com/competitions/ai-mathematical-olympiad-progress-prize-2. Kaggle.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In International Conference on Machine Learning, pages 10835–10866. PMLR.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025a. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. arXiv preprint arXiv:2503.24290.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025b. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. arXiv preprint arXiv:2503.24290.

Xuying Li, Zhuo Li, Yuji Kosuga, and Victor Bian. 2025. Optimizing safe and aligned language generation: A multi-objective grpo approach. arXiv preprint arXiv:2503.21819.

Jie Liu, Gongye Liu, Jiajun Liang, Yangguang Li, Jiaheng Liu, Xintao Wang, Pengfei Wan, Di Zhang, and Wanli Ouyang. 2025a. Flow-grpo: Training flow matching models via online rl. arXiv preprint arXiv:2505.05470.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025b. Understanding r1-zero-like training: A critical perspective. arXiv preprint arXiv:2503.20783.

Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, Maurice Weber, Ce Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025a. Deepcoder: A fully open-source 14b coder at o3-mini level. . Notion Blog.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025b. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. . Notion Blog.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems, 36:53728–53741.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. arXiv preprint arXiv:2409.19256.

Zhiqing Sun. 2024. Aimo-cmu-math/cmu-math-aimo.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. arXiv preprint arXiv:2501.12599.

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, and 1 others. 2025. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond. arXiv preprint arXiv:2503.10460.

Lilian Weng. 2024. Reward hacking in reinforcement learning. lilianweng.github.io.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. arXiv preprint arXiv:2409.12122.

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. 2025. Demystifying long chain-of-thought reasoning in llms. arXiv preprint arXiv:2502.03373.