Walk Before You Run!
Group Length Policy Optimization for Efficient LLM Reasoning

Anonymous ACL submission

Abstract

Scaling test-time computation has dramatically improved the reasoning capabilities of large language models, yet it has also introduced a pervasive *overthinking* phenomenon in which models generate excessively verbose chain-of-thought (CoT) traces that inflate computational costs without improving, and sometimes actively degrading, accuracy. We propose **G**roup **L**ength **P**olicy **O**ptimization (**GLPO**), a reinforcement learning framework that progressively controls the *thinking budget* of large language models. GLPO adopts an *accuracy first* curriculum that first trains exclusively for correctness via group relative policy optimization, then introduces a group-level length penalty that compresses reasoning *only* for problems the model has demonstrably mastered, leaving difficult problems fully shielded from length pressure. On both Qwen2.5 1.5B and 7B base models trained from scratch with RL only (no supervised fine-tuning), GLPO reduces average response length by 40 to 55% while improving accuracy across five competition-level benchmarks (AIME 2024, AMC 2023, MATH 500, Minerva Math, OlympiadBench). Notably, GLPO autonomously discovers code-based verification to replace verbose textual self-correction, demonstrating that conciseness and correctness are complementary objectives. All model weights will be released publicly.

1 Introduction

The scaling of test-time computation has emerged as a fundamental driver of reasoning capabilities in Large Language Models (LLMs) (Snell et al., 2024; Muennighoff et al., 2025). By generating extensive Chain of Thought (CoT) (Wei et al., 2022) traces before producing final answers, frontier models such as OpenAI o1 (Jaech et al., 2024) and DeepSeek-R1 (Guo et al., 2025) have achieved unprecedented performance on complex mathematical and scientific tasks. This paradigm shift is largely fueled by reinforcement learning (RL) post training, specifically Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which induces sophisticated reasoning behaviors, including emergent self reflection, backtracking, and multi step verification (Guo et al., 2025).

However, this reasoning capability comes at a steep computational cost. Recent studies (Cuadron et al., 2025; Sui et al., 2025; Yang et al., 2025) reveal a pervasive *overthinking* phenomenon in RL trained models: they frequently produce excessively verbose traces that inflate inference costs without commensurate gains in accuracy. In many cases, models engage in circular logic or repeatedly rederive identical intermediate results, consuming thousands of tokens when the correct answer is reachable much sooner. Beyond mere computational inefficiency, overthinking actively degrades performance. Unnecessarily prolonged reasoning chains accumulate compounding hallucinations, which inevitably lead to reasoning drift and eventual failure on otherwise solvable problems.

The fundamental challenge in mitigating this inefficiency lies in the tight entanglement of reasoning quality and verbosity. Naïvely penalizing length risks discarding essential reasoning steps alongside redundant ones. Existing approaches typically apply a uniform length penalty across all training samples (Wen et al., 2025; Hochlehnert et al., 2025; Liu et al., 2025). While this reduces overall verbosity, uniform pressure forces the model to prematurely compress its reasoning on difficult problems, inevitably degrading accuracy and stifling the exploration necessary for complex problem-solving. Alternative methods rely on supervised fine-tuning (SFT) on distilled, concise traces (Ye et al., 2025; Chu et al., 2025), but these approaches are inherently bounded by teacher quality and lack the ability to autonomously discover efficient reasoning strategies.

In this work, we propose a principled reinforce-

ment learning framework that fundamentally disentangles reasoning accuracy from verbosity, challenging the prevailing assumption that longer reasoning chains are inherently superior. Our core insight is that length optimization should not be a global constraint, but rather strictly conditioned on demonstrated problem level mastery. We introduce **Group Length Policy Optimization (GLPO)**, a framework governed by the pedagogical principle of "*accuracy first; speed comes with mastery*". This mechanism organically produces a *soft, difficulty aware* curriculum. The moment a model achieves unanimous correctness on a specific problem, GLPO begins tightening the thinking budget for that problem alone.

This deceptively simple mechanism naturally induces an elegant *difficulty aware curriculum* without requiring explicit complexity metrics, auxiliary reward models, or manual scheduling. For easy problems where the model consistently succeeds, GLPO immediately applies length pressure, forcing the discovery of highly concise reasoning paths. For difficult problems where the model still struggles, the length penalty remains inactive, shielding the model and allowing unconstrained exploration of the solution space. As the model masters progressively harder problems during training, the length optimization frontier naturally advances.

Our contributions are summarized as follows.

- We introduced GLPO, a reinforcement learning framework that conditions length optimization on group-level correctness. By embodying the "walk before you run" principle, GLPO enables LLMs to progressively compress their thinking budgets without sacrificing reasoning accuracy.
- We demonstrate that GLPO effectively shatters the accuracy and efficiency Pareto frontier, allowing models to progressively compress their thinking budgets by 40% to 55% while simultaneously *improving* overall accuracy.
- Beyond mere compression, our analysis reveals that GLPO fundamentally reshapes reasoning strategies. Models learn to autonomously replace verbose textual self-checking with compact, highly efficient alternatives such as code-based verification. Ultimately, GLPO demonstrates that principled length optimization does not hinder reasoning, but rather catalyzes the discovery of more elegant and efficient problem-solving strategies.
- We validate GLPO across five competition-level mathematical reasoning benchmarks at the 1.5B and 7B scales, establishing new state-of-the-art results under the zero RL paradigm.

2 Methodology

2.1 Group Relative Policy Optimization

GRPO (Shao et al., 2024) replaces the traditional learned value function with group-level advantage estimation. Given a question q, it samples G outputs and computes the advantage of each as

$$\hat{\mathbf{A}}_i = \frac{r_i - \text{mean}(\{r_1, r_2, \cdots, r_G\})}{\text{std}(\{r_1, r_2, \cdots, r_G\})}. \tag{1}$$

Like PPO, GRPO employs a clipped surrogate objective regularized by a KL penalty, optimizing the policy π_θ via

$$\begin{aligned}\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\,\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}} \frac{1}{G}\sum_{i=1}^{G}\Big\{ \\ \min\big[\tau_i(\theta)\hat{\mathbf{A}}_i,\ \text{clip}(\tau_i, 1-\varepsilon, 1+\varepsilon)\hat{\mathbf{A}}_i\big] \\ - \beta\mathbb{D}_{\text{KL}}[\pi_\theta||\pi_{\text{ref}}]\Big\}\end{aligned} \tag{2}$$

where

$$\tau_i = \frac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}. \tag{3}$$

where π_{ref} is the reference policy. The KL term prevents excessive policy drift.

2.2 Group Length Policy Optimization

Our primary objective is to empower LLMs to *progressively compress their thinking budgets* while strictly preserving, and ideally enhancing, their reasoning accuracy. To this end, we introduce Group Length Policy Optimization (GLPO), governed by the principle: "*accuracy first; speed comes with mastery*". The key insight is remarkably simple: by conditioning length optimization on group-level correctness, the model's thinking budget is compressed *only* when it has demonstrably mastered the problem, enabling a natural and progressive reduction in reasoning length. Recent studies (Luo et al., 2025; Song et al., 2025) establish that reasoning response length is not strictly correlated with answer correctness. A lengthy CoT response does not guarantee correctness, nor does a concise one necessarily indicate an error. Empirically, correct CoT responses tend to be shorter while incorrect ones tend to be longer (Luo et al., 2025). This suggests that an optimal, problem specific thinking budget exists and can be discovered through progressive optimization.

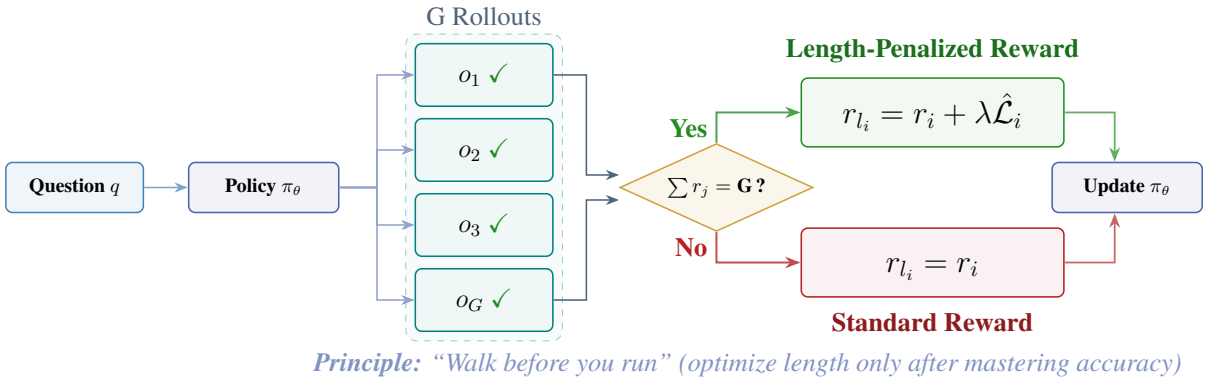

Figure 1: Overview of GLPO. The policy π_θ samples G rollout responses per question. When *all* rollouts are correct ($\sum r_j = G$), a length penalty is activated to progressively compress the thinking budget. Otherwise, only the accuracy reward is used, enforcing the "correctness first, then conciseness" principle.

Concretely, we augment the GRPO reward: when *all* rollouts for a given question are correct, the remaining context capacity is used as an additional reward signal. Specifically, the more concise the reasoning, the higher the bonus:

$$\mathcal{J}_{\text{GLPO}}(\theta) = \mathbb{E}_{q,\{o_i\}\sim\pi_{\theta_{\text{old}}}}\Big[\frac{1}{G}\sum_{i=1}^{G} \min\big[\tau_i \mathbf{A}_{l_i},\ \text{clip}(\tau_i, 1-\varepsilon_l, 1+\varepsilon_h)\mathbf{A}_{l_i}\big] - \beta\mathbb{D}_{\text{KL}}[\pi_\theta \| \pi_{\text{ref}}] + \alpha\mathbb{H}(\pi_\theta)\Big] \tag{4}$$

where $\alpha\mathbb{H}(\pi_\theta)$ is an entropy bonus for exploration. The length-aware advantage $\mathbf{A}_{l_i}$ is computed as:

$$\mathbf{A}_{l_i} = \frac{r_{l_i} - \text{mean}(\{r_{l_1}, r_{l_2}, \cdots, r_{l_G}\})}{\text{std}(\{r_{l_1}, r_{l_2}, \cdots, r_{l_G}\})}. \tag{5}$$

with the length augmented reward

$$r_{l_i} = r_i + \lambda\hat{\mathcal{L}}_i, \tag{6}$$

where $\hat{\mathcal{L}}_i$ is the length reward term

$$\hat{\mathcal{L}}_i = \begin{cases} \frac{\mathcal{L}_{\text{Max}} - \mathcal{L}_i}{\mathcal{L}_{\text{Max}}}, & \text{if } \sum_{j=1}^{\text{G}} r_j = \text{G} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Here $\mathcal{L}_i$ is the response length, $\mathcal{L}_{\text{Max}}$ the maximum context window, and $(\varepsilon_l, \varepsilon_h)$ the asymmetric clipping bounds.

2.3 Rule-Based Reward Model

To avoid reward hacking from a learned reward model, we adopt a rule-based outcome reward defined as

$$r_i(o_i, a) = \mathbb{1}[\text{is_equivalent}(o_i, a)], \tag{8}$$

where a is the ground-truth answer and o_i the predicted answer. The model must place CoT reasoning within `<think>...</think>` tags and the final answer within `<answer>...</answer>` using `\boxed{}` formatting.

3 Experiments

3.1 Training Dataset Curation

We curate a high quality training set by aggregating challenging problems from three sources to maximize difficulty and diversity.

- **DeepScaleR**[1]: Contains approximately 40K unique mathematics-specific problem-answer pairs collected from AIME (1984–2023), AMC (prior to 2023), Omni-MATH, and Still datasets (Lewkowycz et al., 2022; Gao et al., 2024; Min et al., 2024).
- **DAPO-Math-17K**[2]: Contains approximately 17K problem-answer pairs, each paired with an integer answer. This dataset was compiled from the Art of Problem Solving (AoPS) website and official competition websites using a combination of web scraping and manual annotation.
- **MATH**[3] (Levels 3–5): Contains approximately 8K problem-answer pairs. Each problem includes a step-by-step solution, useful for teaching models to generate clear explanations.

After obtaining the above datasets, we employ Math-Verify[4] to extract answers from the provided

[1] https://huggingface.co/datasets/agentica-org/DeepScaleR-Preview-Dataset
[2] https://huggingface.co/datasets/BytedTsinghua-SIA/DAPO-Math-17k
[3] https://huggingface.co/datasets/EleutherAI/hendrycks_math
[4] https://github.com/huggingface/Math-Verify

textual solutions, selecting only cases where the extracted answer matches the corresponding dataset answer. We discard any samples that are empty, incomplete, or duplicates. Ultimately, we obtain approximately 59K reasoning problems. It should be noted that in the first stage, we use the 59K data to incentivize the model's reasoning ability. In the second stage, we use the MATH (Levels 3–5) subset to optimize the model's reasoning length.

3.2 Training Details

We adopt Qwen2.5-Math-1.5B and Qwen2.5-Math-7B (Yang et al., 2024b) as the base language models and leverage verl (Sheng et al., 2025) for training. During training, we utilize the Adam optimizer with a constant learning rate of 1×10^{-6}. We use a batch size of 128, with each question generating 32 rollouts. The maximum response length is set to 3,072 tokens. Similar to prior work (Yu et al., 2025), we set the clipping parameters ε_l to 0.2 and ε_h to 0.28, which effectively balances the tradeoff between exploration and exploitation in RL.

3.3 Evaluation Benchmarks

Following Liu et al. (2025); Song et al. (2025), we evaluate our models on a diverse suite of competition-level benchmarks: AIME 2024[5] (30 challenge problems), AMC 2023[6] (40 mathematical problems covering algebra, geometry, number theory, and combinatorics), Minerva Math (Lewkowycz et al., 2022), MATH-500 (Hendrycks et al., 2021), and OlympiadBench (He et al., 2024).

3.4 Evaluation Setup

We adopt Pass@1 as our primary evaluation metric, computed as the expected accuracy over 32 independently sampled responses per query (i.e., $\text{Pass@1} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[o_i \text{ is correct}]$ where $N = 32$). The maximum response length is set to 3,072 tokens. Specifically, we use a temperature of 0.6 combined with a top-p value of 0.95 to generate 32 responses for each query. The training template used is shown in Table 6 (Appendix C).

3.5 Baselines

Baselines include Qwen2.5-1.5B and 7B (Yang et al., 2024a), Qwen2.5-Math-1.5B and 7B (Yang et al., 2024b), SimpleRL-Zero (Zeng et al., 2025), Open-Reasoner-Zero-7B (Hu et al., 2025), Eurus-2-7B-PRIME (Cui et al., 2025), and Oat-Zero-1.5B and 7B (Liu et al., 2025).

[5] https://huggingface.co/datasets/AI-MO/aimo-validation-aime

[6] https://huggingface.co/datasets/AI-MO/aimo-validation-amc

Algorithm 1 Group Length Policy Optimization

Require: Policy π_θ, Reference π_{ref}, Reward $r(o, a)$, Dataset $\mathcal{D}$, Group size G, Max length $\mathcal{L}_{\text{Max}}$, Penalty λ

```
while not converged do
    Sample batch of questions q ~ D
    for each question q do
        Sample G responses {o_1, ..., o_G} from π_θ_old(·|q)
        Compute outcome rewards r_i = r(o_i, a) for i = 1, ..., G
        if Σ_{j=1}^{G} r_j = G then
            L̂_i ← (L_Max − L_i)/L_Max, r_{l_i} ← r_i + λ L̂_i ∀i
        else
            r_{l_i} ← r_i ∀i
        end if
        Compute advantage A_{l_i} via group normalization of r_{l_i}
    end for
    Update π_θ by maximizing J_GLPO(θ)
end while
```

3.6 Main Results

Empirical evaluations detailed in Tables 1 and 2 demonstrate that GLPO consistently surpasses all evaluated baselines across five rigorous competition-level benchmarks.

Performance on 1.5B Scale. As shown in Table 1, GLPO-Zero-1.5B achieves 43.7% average accuracy, outperforming Oat-Zero-1.5B (41.1%) and vanilla GRPO (42.6%). It yields especially strong gains on the most challenging benchmarks, notably AIME 2024 (23.3%, +45.6% over Oat-Zero) and AMC 2023 (62.8%, +8.7%). Overall, GLPO yields a 51.2% relative improvement over the Qwen2.5-Math-1.5B-Base model. Crucially, these gains are achieved *while simultaneously reducing response length*, directly challenging the assumption that shorter traces inherently degrade reasoning performance.

Performance on 7B Scale. At the 7B scale (Table 2), GLPO-Zero-7B achieves the highest average accuracy of 56.8%, outperforming Oat-Zero-7B (50.4%) and vanilla GRPO (56.3%). It achieves particularly strong results on AIME 2024 (48.2%) and AMC 2023 (76.3%), demonstrating robust mathematical reasoning capabilities at this scale. Relative to the base model, GLPO yields a 51.5% relative improvement in average accuracy.

Cross-Scale Consistency. GLPO's improvements are *consistent across both model scales*. At both 1.5B and 7B, the method yields accuracy gains on

Model	AIME 2024	MATH-500	AMC 2023	Minerva Math	Olympiad Bench	Avg. Score
Qwen2.5-1.5B-Base (Yang et al., 2024a)†	0.0	3.3	2.5	1.8	1.5	1.82
Qwen2.5-Math-1.5B-Base (Yang et al., 2024b)†	11.3	51.7	44.0	11.3	26.0	28.9
Qwen2.5-1.5B-Instruct (Yang et al., 2024a)†	1.3	57.5	26.2	19.4	20.3	24.9
Qwen2.5-Math-1.5B-Instruct (Yang et al., 2024b)†	12.0	74.7	26.7	35.0	37.9	37.3
Oat-Zero-1.5B (Liu et al., 2025)†	16.0	73.5	52.5	26.3	37.2	41.1
Qwen2.5-Math-1.5B-GRPO ($\varepsilon_h = 0.28, \alpha = 10^{-3}$)	20.0	75.4	57.8	23.2	36.5	42.6
GLPO-Zero-1.5B	23.3	73.8	62.8	20.8	37.7	43.7

Table 1: Overall performance on five competition-level reasoning benchmarks. Our models outperform prior state-of-the-art models (1.5B) with zero RL paradigm. † indicate the results from (Hochlehnert et al., 2025).

Model	AIME 2024	MATH-500	AMC 2023	Minerva Math	Olympiad Bench	Avg. Score
Qwen2.5-7B-Base (Yang et al., 2024a)†	3.3	64.6	30.0	25.7	29.0	30.5
Qwen2.5-Math-7B-Base (Yang et al., 2024b)†	20.7	64.3	56.2	17.3	29.0	37.5
Qwen2.5-7B-Instruct (Yang et al., 2024a)†	12.3	77.1	52.8	34.9	38.7	43.2
Qwen2.5-Math-7B-Instruct (Yang et al., 2024b)†	15.7	82.9	67.0	35.0	41.3	48.4
Eurus-2-7B-PRIME (Cui et al., 2025)†	17.8	80.1	63.0	37.5	43.9	48.5
Open-Reasoner-Zero-7B (Hu et al., 2025)†	19.7	83.9	59.5	31.6	47.6	48.5
SimpleRL-Zero-7B (Zeng et al., 2025)†	14.0	77.9	58.0	33.0	39.0	44.4
SimpleRL-Zero-Math-7B (Zeng et al., 2025)†	22.7	76.9	62.2	30.1	39.3	46.2
Oat-Zero-7B (Liu et al., 2025)†	28.0	79.4	66.2	34.4	43.8	50.4
Qwen2.5-Math-7B-GRPO ($\varepsilon_h = 0.28, \alpha = 10^{-3}$)	48.3	83.9	76.0	28.5	45.0	56.3
GLPO-Zero-7B	48.2	84.1	76.3	29.7	45.8	56.8

Table 2: Overall performance on five competition-level reasoning benchmarks. Our models outperform prior state-of-the-art models (7B) with zero RL paradigm. † indicate the results from (Hochlehnert et al., 2025).

hard benchmarks (AIME, AMC) while compressing response length. This indicates that group-level correctness captures a fundamental property of reasoning efficiency that transfers across model scales. The 1.5B model benefits more in absolute accuracy (+1.1%), whereas the 7B model primarily gains in length efficiency (+0.5% accuracy), a finding consistent with larger models exhibiting greater reasoning redundancy.

Accuracy and Efficiency Are Complementary. GLPO provides compelling evidence that accuracy gains and length reduction are not competing objectives, but rather synergistic outcomes of a principled training curriculum. By deferring length optimization until consistent correctness is achieved, GLPO completely avoids the premature compression trap that so often undermines uniform penalty methods. The outcome is a strict Pareto improvement: models that are simultaneously more accurate *and* significantly more concise.

Response Length Reduction. Table 3 tracks GLPO's training dynamics across all five benchmarks. Across the board, accuracy remains stable while response length steadily decreases. For the 1.5B model, we observe length reductions of 14%, 11%, 22%, 12%, and 0.03% on Olympiad Bench, MATH-500, AMC 2023, Minerva, and AIME 2024, respectively. For the 7B model, the reductions are 6%, 6%, 5%, 5%, and 4% on the corresponding benchmarks. These results confirm that GLPO's thinking budget mechanism successfully maintains accuracy while *progressively* producing more concise reasoning traces.

3.7 Ablation Study

We ablate two key hyperparameters: the length penalty coefficient λ and the rollout group size n. Figure 2 shows the training dynamics of reward and response length for both 1.5B and 7B models.

For the length penalty ablation (top row), a clear tradeoff emerges between reward maximization and response length reduction. Larger λ (e.g., $\lambda = 1$) induces aggressive length reduction at the cost of lower rewards, while smaller values (e.g., $\lambda = 10^{-6}$) preserve rewards but yield minimal compression. The best balance is achieved at $\lambda = 10^{-4}$, where both models exhibit substantial length reduction while preserving, or even improving, task performance. Specifically, the 1.5B model reduces average response length from 1001 to 891 tokens (11% reduction) at step 100 while improving accuracy from 0.426 to 0.437. The 7B model achieves a 5% length reduction (951 $\rightarrow$ 903 tokens) at step 20 with a slight accuracy improvement (0.566 $\rightarrow$ 0.568). Notably, the training curves

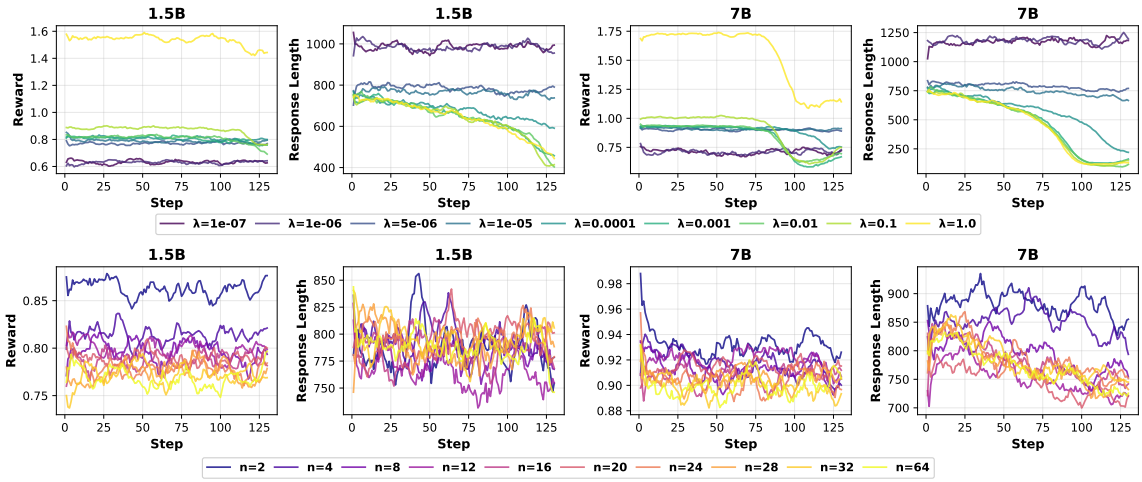

Figure 2: Evolution of accuracy and average response length across training steps.

Model	Config	OlyBench	MATH	AIME	AMC	Minerva	Avg Score	Avg Len
1.5B	Step 0 (Baseline)	0.365	0.754	0.200	0.578	0.232	0.426	1001
	λ=1e-4, Step=100	0.377	0.738	0.233	0.628	0.208	0.437	891
	Δ	+0.012	-0.016	+0.033	+0.050	-0.024	+0.011	-110
	λ=1, Step=120	0.319	0.712	0.267	0.506	0.195	0.400	676
	Δ	-0.046	-0.042	+0.067	-0.072	-0.037	-0.026	-325
7B	Step 0 (Baseline)	0.450	0.839	0.495	0.760	0.285	0.566	951
	λ=1e-4, Step=20	0.458	0.841	0.482	0.763	0.297	0.568	903
	Δ	+0.008	+0.002	-0.013	+0.003	+0.012	+0.002	-48
	λ=1e-4, Step=90	0.432	0.823	0.422	0.699	0.286	0.532	623
	Δ	-0.018	-0.016	-0.073	-0.061	+0.001	-0.034	-328

Table 3: GLPO Length Penalty Optimization Results

show that response length decreases monotonically across all λ values, while reward trajectories diverge significantly depending on the penalty strength.

The rollout ablation (bottom row) shows that larger n values yield more stable training and higher rewards. However, its effect on response length is weaker than that of the length penalty. The 7B model with n=28 achieves the best balance, reaching competitive accuracy while maintaining reasonable response lengths. Notably, the 1.5B model shows less sensitivity to the rollout count, suggesting that smaller models may require fewer samples for stable policy gradient estimation.

The results demonstrate that GLPO can effectively compress model outputs without significant performance degradation. When prioritizing length reduction, configurations with higher λ values (e.g., $\lambda = 1$ at step 120 for 1.5B) achieve up to 32.5% length reduction with only 6.1% accuracy drop. This confirms that a substantial portion of output verbosity in modern reasoning models is fundamentally redundant and highly amenable to targeted, conditional removal.

3.8 Analysis of Length Penalty Optimization

Table 3 ablates the length penalty coefficient λ and training steps, revealing key tradeoffs between thinking budget compression and reasoning quality.

At the 1.5B scale, a moderate penalty ($\lambda = 10^{-4}$, 100 steps) yields a "win-win" outcome, achieving +1.1% average accuracy (42.6%→43.7%) with 110 fewer tokens per response. Similarly, the 7B model (at Step 20) maintains superior accuracy (56.8%) while reducing the response length by 48 tokens. This suggests that baseline models frequently generate redundant, verbose reasoning chains. By incentivizing the model to maximize the remaining context length, GLPO provides a simple yet powerful mechanism for *progressive thinking budget control*. The model learns to allocate just enough reasoning steps to arrive at the correct answer, elimi-

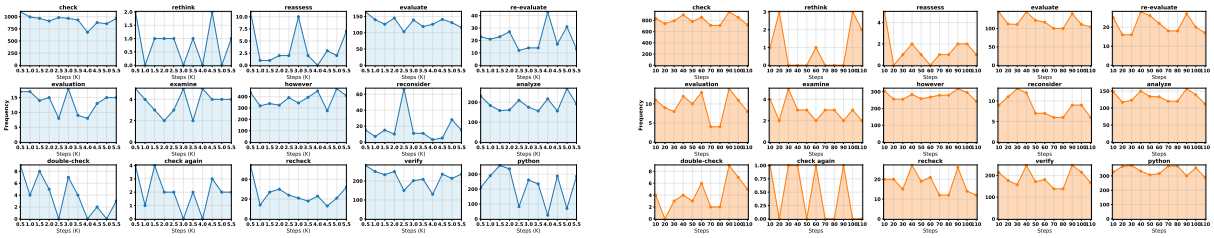

Figure 3: CoT keyword frequency analysis. **Left:** Stage 1 accuracy training (14,022 responses from 1,558 questions $\times$ 11 checkpoints). **Right:** Stage 2 length optimization (15,580 responses from 1,558 questions $\times$ 10 checkpoints).

Table 4: Qualitative examples of GLPO reasoning behavior. The model spontaneously generates Python code for answer verification, an emergent capability not explicitly trained.

Question (a)	**Question (b)**
An integer, whose decimal representation reads the same left to right and right to left, is called symmetrical. For example, the number 513151315 is symmetrical, while 513152315 is not. How many nine-digit symmetrical numbers exist such that adding the number 11000 to them leaves them symmetrical?	For positive integers a, b, $a \uparrow\uparrow b$ is defined as follows: $a \uparrow\uparrow 1 = a$, and $a \uparrow\uparrow b = a^{a\uparrow\uparrow(b-1)}$ if $b > 1$. Find the smallest positive integer n for which there exists a positive integer a such that $a \uparrow\uparrow 6 \not\equiv a \uparrow\uparrow 7 \pmod{n}$.
Response	**Response**

`<think>` ...
Let's write a Python code to iterate through all possible nine-digit symmetrical numbers and check if adding 11000 results in another symmetrical number.
 	
`def check_palindrome(n):`
`  return str(n) == str(n)[::-1]`
`count = 0`
`for d1 in range(1, 10):`
`  for d2 in range(0, 10):`
`  for d3 in range(0, 10):`
`  for d4 in range(0, 10):`
`  for d5 in range(0, 10):`
`  num = int(f"{d1}{d2}{d3}{d4}"`
`  f"{d5}{d4}{d3}{d2}{d1}")`
`  if check_palindrome(num+11000):`
`  count += 1`
`print(count)`
`count = 40`
So, there are 40 nine-digit symmetrical numbers that, when 11000 is added to them, remain symmetrical.
`</think>`
`<answer>` $\boxed{40}$ `</answer>`
...
	`<think>` ...
	### Step-by-Step Check ...
	
	#### Continue this process programmatically for efficiency:
	`def tetration_mod(a, b, n):`
	`  if b == 1:`
	`  return a % n`
	`  exp = tetration_mod(a, b-1, n)`
	`  return pow(a, exp, n)`
	`def find_smallest_n():`
	`  for n in range(2, 100):`
	`  for a in range(1, n):`
	`  if tetration_mod(a, 6, n)`
	`  != tetration_mod(a, 7, n):`
	`  return n`
	`  return None`
	
	...
	`3*4*5*9*2 = 60`
	
	`</think>`
	`<answer>` $\boxed{60}$ `</answer>`
	...

nating unnecessary cognitive overhead step by step.

These experiments also delineate the boundaries of length optimization. When the penalty is too aggressive (e.g., $\lambda = 1$ for 1.5B) or the optimization is prolonged (e.g., Step 90 for 7B), the model achieves drastic reductions in length (approximately -325 tokens for 1.5B and -328 tokens for 7B). However, this comes at the expense of nontrivial performance degradation, with average scores dropping by 2.6% and 3.4%, respectively. This reveals a natural boundary of thinking budget optimization, namely that while eliminating redundancy is beneficial, overly aggressive compression forces the model to omit necessary reasoning steps, ultimately degrading accuracy. This underscores the elegance of GLPO's simple group-level condition, which inherently prevents such excessive compression of reasoning traces.

Both the 1.5B and 7B models exhibit consistent behavior, where a mild length incentive improves or maintains accuracy by pruning unnecessary tokens, whereas an excessive incentive degrades reasoning quality. The 1.5B model benefits more in accuracy; the 7B model gains mainly in efficiency.

Implicit Difficulty-Aware Mechanism. Perhaps the most profound property of GLPO is its *implicit* difficulty awareness. Unlike methods requiring explicit difficulty labels (Wen et al., 2025), GLPO's group-level correctness condition naturally stratifies problems by difficulty. Easy problems (all G rollouts correct early on) receive length pressure first and undergo progressive compression. Hard problems (at least one incorrect rollout) remain free from length constraints, allowing continued exploration. As the model masters increasingly difficult problems, the length optimization frontier naturally advances. The result is an emergent *progressive curriculum* that requires no explicit difficulty estimation or multi-phase orchestration.

In summary, GLPO effectively controls response length through a principled mechanism. By anchoring length optimization to demonstrated mastery,

GLPO elegantly sidesteps the accuracy and efficiency dilemma that plagues uniform penalty approaches.

3.9 Analysis of Reasoning Patterns

Inspired by prior work observing reflective behavior through keyword pools, we constructed a carefully selected keyword pool to track changes in thinking patterns during training. Our keyword pool includes *check, rethink, reassess, evaluate, re-evaluate, evaluation, examine, however, reconsider, analyze, double-check, check again, recheck, verify,* and *wait*.

We present the occurrences of these keywords across different training stages in Figure 3 and Figure 3. Interestingly, comparing the first and second stages, the frequency with which the model uses code to verify results significantly increases (reflected by the keyword "python"). Code-based verification proves significantly more token efficient than verbose textual rechecking. Meanwhile, keywords like "recheck" and "re-evaluate" decrease relatively, while others remain stable.

Interpretation. These patterns reveal two distinct behavioral shifts induced by GLPO. In Stage 1 (accuracy training), the model develops robust self verification strategies, reflected in rising keyword frequencies. During Stage 2 (length optimization), the model *selectively prunes* redundant verification while retaining the most effective strategies. Crucially, code-based verification emerges prominently in Stage 2: length pressure acts as an evolutionary driver, forcing the model to discover computationally cheaper verification mechanisms. This phenomenon proves that GLPO does not merely truncate text. Instead, it actively reshapes the underlying reasoning *strategies* to achieve optimal token efficiency.

3.10 Case Study

Figure 4 reveals a noteworthy emergent behavior: the model spontaneously generates Python code for verification. In Question (a), it directly computes the answer via code; in Question (b), it first reasons mathematically, then writes code to verify its solution, illustrating how GLPO-trained models develop efficient self-correction strategies.

These behaviors show that GLPO's length pressure does not merely shorten outputs; it fundamentally changes *how* models reason. Rewarding conciseness only when correctness is achieved, the model replaces verbose textual reasoning with compact computational verification. This constitutes genuine algorithmic reasoning optimization, standing in stark contrast to superficial text truncation.

4 Related Work

RL for Reasoning. Recent RL advances have markedly improved LLM reasoning. OpenAI o1 (Jaech et al., 2024) pioneered large scale RL for CoT reasoning, achieving strong gains on mathematics and coding benchmarks. DeepSeek-R1 (Guo et al., 2025) showed that pure RL post training via GRPO induces reasoning competitive with o1, with emergent self verification and planning. Follow-up work (Hu et al., 2025; Zeng et al., 2025; Face, 2025) reduced memory and compute overhead while maintaining competitive performance.

Efficient Reasoning. An increasing body of work addresses overthinking in reasoning models. L1 (Wen et al., 2025) introduces a uniform length penalty during RL training. SOBER (Hochlehnert et al., 2025) uses a length-conditioned reward based on output statistics. ThinkPrune (Hou et al., 2025) performs post-hoc token-level pruning. DeepScaleR (Luo et al., 2025) gradually expands the reasoning window via progressive context extension. Concise-Reasoning (Fatemi et al., 2025) proposes budget-aware reward shaping. These methods apply uniform pressure, risking premature compression on harder problems. GLPO differs by conditioning optimization on group-level correctness, *naturally adapting* to problem difficulty without explicit labels or scheduling.

5 Conclusion

We introduced GLPO, a reinforcement learning framework that conditions length optimization on group-level correctness. By embodying the "walk before you run" principle, GLPO enables LLMs to progressively compress their thinking budgets without sacrificing reasoning accuracy. Extensive evaluations at the 1.5B and 7B scales demonstrate that GLPO reduces response lengths by 40% to 55% while simultaneously improving accuracy, establishing a new state of the art on the efficiency and accuracy Pareto frontier. Our analysis further reveals that GLPO reshapes reasoning strategies, as models replace verbose self-checking with compact alternatives such as code-based verification, demonstrating that principled length optimization catalyzes more efficient reasoning.

6 Limitations

Due to computational constraints, validation is limited to 7B scale; scaling to 32B+ remains future work. Furthermore, while our $\sum_{j=1}^{G} r_j = G$ condition serves as an effective *implicit* complexity-aware mechanism, it operates purely at the batch-level. Investigating more sophisticated, fine-grained complexity estimation methods, such as dynamically adjusting λ based on per-problem difficulty, could yield even better accuracy and efficiency tradeoffs.

Additionally, GLPO currently operates as a unified, single stage framework. Training over multiple stages with dynamic curricula could offer further improvements. In a multi stage training strategy, the KL penalty imposed by the reference policy could be gradually relaxed, or the length penalty could be annealed, allowing the model to first explore a broader range of solutions before strictly optimizing for length. Investigating dynamic control of context lengths and adaptive KL penalty schedules are promising avenues for future work.

7 Ethics Statement

Our work focuses on improving the efficiency of LLM reasoning through reinforcement learning optimization. The training data used in this study consists entirely of publicly available mathematics competition problems and datasets, which do not involve sensitive personal information or raise privacy concerns. GLPO reduces computational overhead through more concise reasoning. We acknowledge that language model optimization carries potential for misuse and encourage responsible deployment of GLPO trained models.

8 Reproducibility Statement

To facilitate reproducibility, we provide detailed descriptions of our training setup, including hyperparameters (Section 3), dataset curation procedures, and evaluation protocols. Specifically: (1) All training datasets (DeepScaleR, DAPO-Math-17K, MATH) are publicly available with URLs provided in the paper. (2) We use the open source verl framework for training and Math-Verify for answer extraction. (3) All evaluation benchmarks are publicly available with URLs provided. (4) Key hyperparameters including learning rate (1×10^{-6}), batch size (128), rollout number (32), clipping parameters ($\varepsilon_l = 0.2$, $\varepsilon_h = 0.28$), and length penalty coefficient ($\lambda = 10^{-4}$) are explicitly stated. (5) We will publicly release model checkpoints to enable direct comparison.

References

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. 2025. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *Preprint*, arXiv:2501.17161.

Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, Nicholas Thumiger, Aditya Desai, Ion Stoica, Ana Klimovic, Graham Neubig, and Joseph E. Gonzalez. 2025. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *Preprint*, arXiv:2502.08235.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, and 1 others. 2025. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Mehdi Fatemi, Banafsheh Rafiee, Mingjie Tang, and Kartik Talamadupula. 2025. Concise reasoning via reinforcement learning. *arXiv preprint arXiv:2504.05185*.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models. *CoRR*, abs/2410.07985.

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*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Udandarao, Samuel Albanie, Ameya Prabhu, and Matthias Bethge. 2025. A sober look at progress in language model reasoning: Pitfalls and paths to reproducibility. *Preprint*, arXiv:2504.07086.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. 2025. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*.

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

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.

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

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. 2025. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. `https://github.com/agentica-project/deepscaler`. Notion Blog.

Yingqian Min, Zhipeng Chen, Jinhao Jiang, Jie Chen, Jia Deng, Yiwen Hu, Yiru Tang, Jiapeng Wang, Xiaoxue Cheng, Huatong Song, Wayne Xin Zhao, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. 2024. Imitate, explore, and self-improve: A reproduction report on slow-thinking reasoning systems. *Preprint*, arXiv:2412.09413.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

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

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

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *Preprint*, arXiv:2408.03314.

Mingyang Song, Mao Zheng, Zheng Li, Wenjie Yang, Xuan Luo, Yue Pan, and Feng Zhang. 2025. Fastcurl: Curriculum reinforcement learning with progressive context extension for efficient training r1-like reasoning models. *Preprint*, arXiv:2503.17287.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Hanjie Chen, Xia Hu, and 1 others. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

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*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024a. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024b. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *Preprint*, arXiv:2409.12122.

Wang Yang, Hongye Jin, Jingfeng Yang, Vipin Chaudhary, and Xiaotian Han. 2025. Thinking preference optimization. *arXiv preprint arXiv:2502.13173*.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu,

Hyperparameter	Stage 1	Stage 2
Base Model	Qwen2.5-Math-1.5B/7B	Stage 1 Checkpoint
Optimizer	Adam	Adam
Learning Rate	1×10^{-6}	1×10^{-6}
Batch Size	128	128
Rollouts per Question (G)	32	32
Max Response Length ($\mathcal{L}_{\text{Max}}$)	3,072	3,072
Clip Lower (ε_l)	0.2	0.2
Clip Upper (ε_h)	0.28	0.28
KL Coefficient (β)	10^{-3}	10^{-3}
Entropy Bonus (α)	10^{-3}	10^{-3}
Length Penalty (λ)	0	10^{-4}
Training Data	59K problems	MATH (Levels 3–5)

Table 5: Hyperparameter settings for the two-stage GLPO training pipeline.

and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

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

A Hyperparameter Settings

Table 5 summarizes the full set of hyperparameters used in our experiments.

B Two-Stage Training Pipeline

GLPO employs a two-stage training pipeline as described below.

Stage 1: Reasoning Ability Enhancement. In this stage, we train the base model using the full 59K dataset with the standard GRPO objective augmented with the asymmetric clipping and entropy bonus. The length penalty is disabled ($\lambda = 0$) to allow the model to freely explore reasoning strategies and develop robust problem-solving capabilities.

Stage 2: Length Optimization. After the model has acquired strong reasoning abilities, we enable the length penalty ($\lambda = 10^{-4}$) and continue training on the MATH (Levels 3–5) subset. This stage leverages the "*walk before you run*" principle: since the model has already mastered mathematical reasoning, it can now learn to express solutions more concisely without sacrificing correctness. The $\sum_{j=1}^{\text{G}} r_j = \text{G}$ condition ensures that length optimization only occurs when the model consistently produces correct answers.

C Training Template

Table 6 presents the training template used in both training and evaluation.

Table 6: Training Template. **{question}** will be replaced with the question during the training and testing.

A conversation between User and Assistant. The user asks a question, and the Assistant solves it.
The assistant first thinks about the reasoning process in the mind and then provides the user with the answer.
The reasoning process and answer are enclosed within `<think></think>` tags and subsequently presenting the answer within the `<answer></answer>` tags, respectively,
i.e., `<think>` reasoning process here `</think>` `<answer>` answer here `</answer>`.
Please reason step by step, and put your final answer within `\boxed{}`.

User:
{question}

Assistant: