

GENERATIVE ADVERSARIAL OPTIMIZATION: DUAL-REWARD REINFORCEMENT LEARNING FOR MATHEMATICS REASONING

Anonymous authors

Paper under double-blind review

ABSTRACT

Despite recent progress achieved by large language models (LLMs), their remarkable mathematics reasoning abilities are largely dependent on fine-tuning on the annotated data, lacking generalization on out-of-distribution tasks. To address this, current methods adopt reinforcement learning (RL) to incentivize the latent capabilities of LLMs, mitigating the need for annotations. However, they often suffer from uncontrollable data difficulty and limited initial capabilities. In this paper, we propose **Generative Adversarial Optimization (GAO)**, a novel reinforcement learning framework consists of a problem poser and a problem solver which are optimized by dual-reward iteratively. Specifically, the poser attempts to propose challenging problems to stump the solver, while the solver strives to solve them. The complete adversarial process is recorded to generate bidirectional rewards, enabling both the poser and solver to co-evolve through this competitive interaction. Experimental results show that **GAO** achieves state-of-the-art performance compared to previous models of the same size, even without relying on proprietary LLMs.

1 INTRODUCTION

Recent large language models (LLMs) have demonstrated impressive performance on reasoning tasks (Wang et al., 2023; Taylor et al., 2022). These successes highlight that fine-tuning on vast amounts of annotated data significantly enhances their core reasoning abilities. However, the effectiveness of post-training is heavily dependent on the availability of high-quality annotated data (Yu et al., 2024; Yuan et al., 2023; Luo et al., 2025; Li et al., 2024), and challenges of data collection and annotation remain difficult to overcome (Feng et al., 2025). Given that, some approaches like Qwen3 (Yang et al., 2025), Seed-Thinking-v1.5 (Yu et al., 2025b), Light-R1 (Wen et al., 2025), and DeepSeek-R1 (DeepSeek-AI et al., 2025), adopt reinforcement learning (RL) to mitigate the need for annotations. They encourage LLMs to explore the output space and use rewards to reinforce or penalize the behavior of LLMs, leading to better generalization (Chu et al., 2025). However, the difficulty of training data is uncontrollable, which makes the effect of RL hard to guarantee. Too complex or too easy inputs usually contribute to highly skewed reward distribution (Team et al., 2025), significantly diminishing learning efficiency. Besides, the initial capabilities of LLMs can also affect the ceiling of RL (AI et al., 2025). The capabilities of a model trained through reinforcement learning do not grow indefinitely (Gandhi et al., 2025). For certain challenging problems, LLMs often fails to find the optimal solution even after multiple explorations (Yue et al., 2025), leading to the failure of RL, and thus the performance of models may stagnate.

The challenges mentioned above motivated us to propose **Generative Adversarial Optimization (GAO)**, which enhances the reasoning abilities of LLMs through dual-reward reinforcement learning. As illustrated in Figure 1, SFT trains LLMs to memorize the training data, often struggling to generalize to out-of-distribution tasks, while RL employs reward models (or reward functions) to provide feedback during the exploration process of the model, yet still performs poorly in difficult scenarios having a gap with the model’s capabilities as the exploration in such areas is often insufficient. In contrast, **GAO** progressively enhances the model’s capabilities by continuously identifying and targeting its weaknesses. Specifically, we first utilize a problem poser to construct challenging questions which the problem solver (i. e. the target reasoning model) can hardly answer, finding the

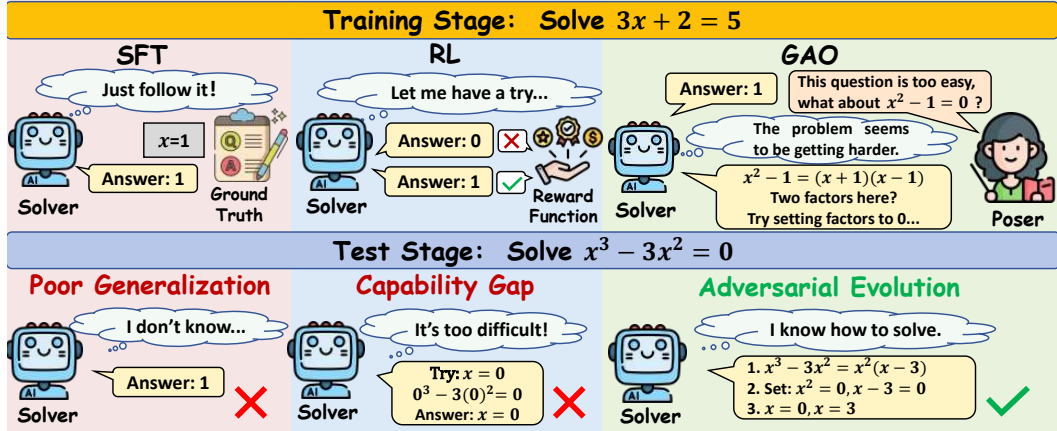


Figure 1: The comparisons between our method and previous training methods. SFT trains LLMs to fit the training data, contributing to the poor generalization. Traditional reinforcement learning methods (e.g., PPO, GRPO, and so on) encourage LLMs to explore different solutions and utilize reward functions to provide feedback, leading to failure when the problems exceed the models’ capabilities. In contrast, GAO progressively enhances the model’s capabilities by continuously identifying and targeting its weaknesses.

tasks where the target model can obtain more gains. Then we conduct RL for the solver on these questions with consistency voting from external reasoning experts. Meanwhile, we also train and update the poser with the reward signal derived from solver’s performance on generated questions. The process of **GAO** mentioned above proceeds iteratively. Through this process, the poser becomes increasingly tricky, while the solver grows progressively strong and robust. The main contributions of this paper are summarized as follows:

- We identify the limitations of current post-training methods (SFT and RL) and design a novel training paradigm to overcome them.
- We propose **Generative Adversarial Optimization (GAO)**, which consists of a poser and a solver and enhances the reasoning abilities of LLMs through dual-reward reinforcement learning. Compared to the previous methods on the backbones of the same size, **GAO** improves the ceiling of the post-training, achieving state-of-the-art (SOTA) performance.
- Extensive experiments demonstrate the effect of **GAO** on multiple reasoning-related tasks, with ablation and analysis studies explaining how and why it works.

2 RELATED WORK

2.1 REASONING LARGE LANGUAGE MODEL

Reasoning plays a crucial role in intellectual activities of LLMs (Huang & Chang, 2023), attracting significant interest from both academia and industry. With the increase of the size, recent LLMs have made significant advancements in a wide range of reasoning tasks such as arithmetic, commonsense, symbolic reasoning (Qiao et al., 2023). Despite such impressive progress, the performance of current open-source reasoning LLMs (Wen et al., 2025; DeepSeek-AI et al., 2025; Yang et al., 2025) still lags behind that of proprietary ones (e.g., o3, o4-mini, gemini-2.5-pro, gpt-4.5-preview, etc. (Anil et al., 2023; Ouyang et al., 2022; OpenAI et al., 2024)) (Chiang et al., 2024), primarily because stronger models often keep their training data proprietary. As a result, the lack of publicly available reasoning datasets remains a significant barrier to further development in this field.

2.2 GENERATIVE ADVERSARIAL NETWORK

Generative Adversarial Network (GAN) is a deep network designed for generating synthetic data that mimics real data distributions. Introduced by Goodfellow et al. (2020), GAN consists of a gen-

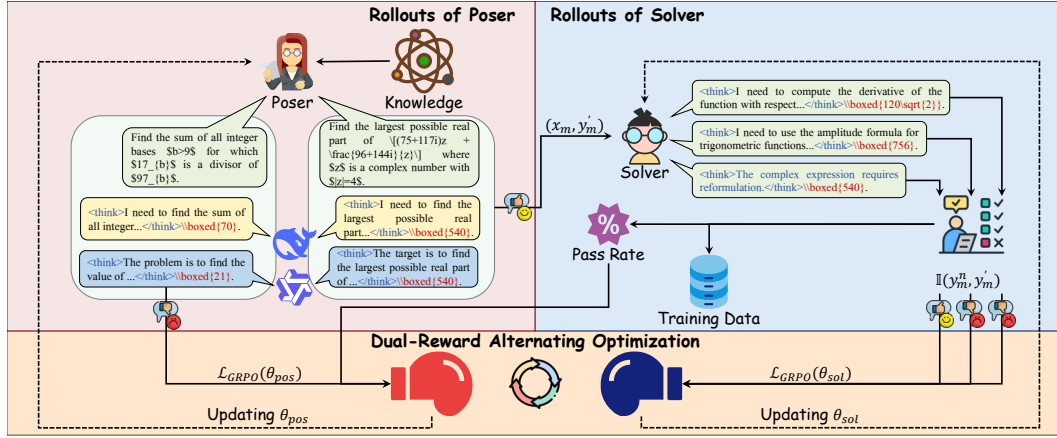


Figure 2: Illustration of the GAO training paradigm. At each turn, the Poser samples candidate problems, which are filtered through a consistency check before being passed to the Solver. The Solver’s pass rate not only guides its own training but also provides feedback for the Poser to generate increasingly challenging problems.

erator creating fake data samples and a discriminator distinguishing between real and fake samples. Through such adversarial training (Gulrajani et al., 2017), GAN achieves high-fidelity data generation across various domains, including images, text, and audio. Inspired by GANs, recent works in code generation and mathematical reasoning, such as Absolute-Zero-Reasoner (Zhao et al., 2025) and SvS (Liang et al., 2025), enable self-play by applying both problem-generation and problem-solving objectives to a single model. However, this design leads to unstable training, and a more effective solution still needs to be explored.

2.3 DATA FILTERING FOR REINFORCEMENT LEARNING

The quality of training data is crucial for the final performance of Reinforcement Learning. The problems like noisy or sparse rewards (Hare, 2019), shifted distributions (Agarwal et al., 2021), exploration bottlenecks (Bai et al., 2021), and adversarial trajectories (Pinto et al., 2017) can significantly affect its effectiveness. Therefore, the process of selectively curating or preprocessing training data to enhance the efficiency and robustness of RL sampling, bridging gaps between simulated and real-world deployment (Kumar et al., 2020).

3 METHODOLOGY

In this section, we describe how **GAO** progressively enhances the model’s capabilities through dual-reward reinforcement learning. Specifically, we first initialize the poser and solver with SFT datasets, then we optimize both of them in a generative adversarial way. The training process of the poser and solver iteratively and alternately proceeds until no more gain can be obtained. It is worth noting that we treat the output of their interaction as the training trajectory of **GAO**, replacing model updating with data updating to avoid severe shifted distribution issues.

3.1 INITIALIZATION FOR THE POSER AND SOLVER

The basic capabilities of the models are crucial for subsequent reinforcement learning. Given that, we initialize the solver by fine-tuning it on the open-source SFT dataset - DeepMath-103K (He et al., 2025). The training objective can be formulated as:

$$\mathcal{L}_{\text{SFT}}(\theta_{\text{sol}}) = - \sum \log P_{\theta_{\text{sol}}}(y^* | x) \quad (1)$$

where θ_{sol} indicates the parameter of the solver. x and y^* are the input (question) and gold output (solution).

As for the poser, we hope it can design various questions based on different knowledge points and ultimately construct high-quality, diverse data. Inspired by PromptCoT (Yao et al., 2024), we utilize Qwen2.5-72B-Instruct (Team, 2024) to summarize knowledge points k which are used by the poser to generate questions (see Appendix A for details). After collecting k , we can initialize the poser:

$$\mathcal{L}_{\text{SFT}}(\theta_{\text{pos}}) = - \sum \log P_{\theta_{\text{pos}}}(c, x \mid k) \quad (2)$$

where θ_{pos} indicates the parameter of the poser. c can be regarded as the Chain-of-Thoughts before formulating the final questions.

3.2 TRAINING FOR THE POSER

Each training iteration of **GAO** starts with the poser’s training phase. The poser is expected to identify the weaknesses of the solver, so we use the failure rates of the solver on the synthetic problems generated by the poser as the rewards to optimize it. Specifically, we first utilize the poser to sample M questions $x_{1:M}$ based on each knowledge point k . Then we employ strong reasoning models (Qwen3-235B-A22B (Yang et al., 2025) and DeepSeek-R1 (DeepSeek-AI et al., 2025)) to conduct consistency voting, getting the top-voted answer y'_m for each question x_m . Given that, we can get the reward function for the poser:

$$\begin{aligned} x_1, x_2, \dots, x_M &\sim \pi_{\theta_{\text{pos}}}(k) \quad y_m^1, y_m^2, \dots, y_m^N \sim \pi_{\theta_{\text{sol}}}(x_m) \\ r_m^{\text{pos}} &= \begin{cases} 1 - \frac{1}{N} \sum_{n=1}^N \mathbb{I}(y_m^n = y'_m) & \exists n, \mathbb{I}(y_m^n, y'_m) = 1 \\ 0 & \forall n, \mathbb{I}(y_m^n, y'_m) = 0 \end{cases} \end{aligned} \quad (3)$$

where N is the number of rollouts of the solver and $\mathbb{I}$ is the rule-based judge function. y_m^n is the n -th output sampled from the solver fed with the question x_m .

From the perspective of the overall training objectives, the poser serves as an indicator, highlighting the gap between the solver and the current state-of-the-art reasoning model. In particular, when these strong models fail to reach a consensus or the solver is completely unable to correctly answer a question posed by the poser, we consider that the question is overly difficult or potentially unsolvable. Such problems provide no benefit, or may even be harmful, to the subsequent RL for the solver. Therefore, we discourage the poser from generating such questions and set their reward to 0. Finally, we employ GRPO to optimize the poser. Inspired by DAPO (Yu et al., 2025a) and Dr.GRPO (Liu et al., 2025), we remove the KL term and adopt token-level average pooling as the loss aggregating mode:

$$\begin{aligned} \mathcal{L}_{\text{GRPO}}(\theta_{\text{pos}}) &= \mathbb{E}_{k \sim \mathcal{K}, x \sim \pi_{\theta_{\text{pos}}}(\cdot \mid k)} \frac{1}{\sum_{m=1}^M |x_m|} \\ &\quad \left[\sum_{m=1}^M \sum_{t=1}^{|x_m|} (\min(z_m^t A_m^t, \text{clip}(z_m^t, 1 - \varepsilon, 1 + \varepsilon) A_m^t)) \right] \\ z_m^t &= \frac{\pi_{\theta_{\text{pos}}}(x_{m,t} \mid k, x_{m,<t})}{\pi_{\theta_{\text{pos}}^{\text{old}}}(x_{m,t} \mid k, x_{m,<t})}, A_m^t = \frac{r_m^{\text{pos}} - \text{avg}(r_{1:M}^{\text{pos}})}{\text{std}(r_{1:M}^{\text{pos}})} \end{aligned} \quad (4)$$

3.3 TRAINING FOR THE SOLVER

The training process for the solver is conducted on the poser-generated problems. Similar to the sampling process during the training stage of the poser, we can collect problems with top-voted answers y'_m . Given that, we can obtain the reward function for the solver:

$$r_n^{\text{sol}} = \mathbb{I}(y_m^n, y'_m) \quad (5)$$

Before the prior to the formal training phase, we conduct problems filtering to select samples with a pass rate between P_{low} and P_{high} . Such data presents an appropriate level of challenge to the solver’s reasoning capabilities, which yields greater benefits during reinforcement learning. On the other hand, the filtering process can significantly reduce the volume of training data, thereby greatly improving training efficiency.

Based on the filtered data, we fine-tune the solver’s policy to maximize rewards, encouraging it to make precise, step-by-step logical deductions. The training objective is defined as:

$$\begin{aligned} \mathcal{L}_{\text{GRPO}}(\theta_{\text{sol}}) &= \mathbb{E}_{x_m \sim \pi_{\theta_{\text{pos}}}, y_m^n \sim \pi_{\theta_{\text{sol}}}(\cdot | x_m)} \frac{1}{\sum_{n=1}^N |y_m^n|} \\ &\quad \left[\sum_{n=1}^N \sum_{t=1}^{|y_m^n|} (\min(z_n^t A_n^t, \text{clip}(z_n^t, 1 - \varepsilon, 1 + \varepsilon) A_n^t)) \right] \\ z_n^t &= \frac{\pi_{\theta_{\text{sol}}}(y_m^{n,t} | y_m^{n,<t})}{\pi_{\theta_{\text{sol}}^{\text{old}}}(y_m^{n,t} | y_m^{n,<t})}, A_n^t = \frac{r_n^{\text{sol}} - \text{avg}(r_{1:N}^{\text{sol}})}{\text{std}(r_{1:N}^{\text{sol}})} \end{aligned} \quad (6)$$

3.4 DUAL-REWARD ALTERNATING OPTIMIZATION

To ensure the training stability, GAO alternately optimizes the poser and solver, enabling the target model to focus more on the current task in each optimization step. This training approach is inspired by GAN: the poser acts as the generator, striving to identify capability gaps between the solver and SOTA mathematics reasoning LLMs while posing problems; the solver serves as the discriminator, aiming to solve the proposed problems accurately without being stumped by the poser.

Algorithm 1 GAO Algorithm

Input: Poser θ_{pos}^0 , Solver θ_{sol}^0 , Initial SFT Datasets D_{pos} and D_{sol}

Output: Solver θ_{sol}

- 1: Initialize the parameters of Poser and Solver on D_{pos} and D_{sol} by $\mathcal{L}_{\text{SFT}}$: $\theta_{\text{pos}}^0, \theta_{\text{sol}}^0$
 - 2: Initialize the RL training buffer $B \leftarrow \emptyset$
 - 3: **for** $i = 0$ **to** T **do**
 - 4: Sample problems from the poser based on the given knowledge points k : $\{x_m\}_{m=1}^M \sim \pi_{\theta_{\text{pos}}^i}$
 - 5: Employ strong reasoning models to get the top-voted answer y_m'
 - 6: Sample trajectories from the solver for each x_m : $\{y_m^n\}_{n=1}^N \sim \pi_{\theta_{\text{sol}}^i}$
 - 7: Calculate r_{pos}^m in accordance with Equation 3 and then train the poser by $\mathcal{L}_{\text{GRPO}}(\theta_{\text{pos}})$:
 $\theta_{\text{pos}}^{i+1} \leftarrow \theta_{\text{pos}}^i$
 - 8: **for** $m = 1$ **to** M **do**
 - 9: Calculate pass rate p_m and $\{r_m^n\}_{n=1}^N$ for $\{y_m^n\}_{n=1}^N$
 - 10: **if** $P_{\text{low}} \leq p_m \leq P_{\text{high}}$ **then**
 - 11: Append $\{x_m, (y_m^1, r_m^1), \dots, (y_m^N, r_m^N)\}$ to B
 - 12: **end if**
 - 13: **end for**
 - 14: Update the solver by $\mathcal{L}_{\text{GRPO}}(\theta_{\text{sol}})$: $\theta_{\text{sol}}^{i+1} \leftarrow \theta_{\text{sol}}^i$ using the buffer B
 - 15: **end for**
 - 16: **return** θ_{sol}^T
-

As shown in Algorithm 1, GAO alternately updates the poser and solver in an adversarial loop: the poser generates progressively harder problems that expose the solver’s weaknesses, while the solver improves by attempting to solve them. The solver’s success rate is used as feedback to push the poser to produce harder, more informative tasks. Through this competition, posing and solving improve together.

4 EXPERIMENTS

4.1 SETTINGS

Models and Datasets We adopt Qwen3-8B-Base (Yang et al., 2025) as the backbone for the solver and Qwen3-8B as the backbone for the poser to evaluate the effectiveness of our method. We use the open-source mathematical dataset DeepMath-103K (He et al., 2025) as the SFT dataset for initializing the solver. After applying simple deduplication, we use the resulting SFT dataset

to perform the initialization training of the solver. For the poser initialization, we randomly sample problems from the above datasets and generate SFT data using Qwen2.5-72B-Instruct (Team, 2024), with the prompts detailed in Appendix A.

Implementation Details We employ verl (Sheng et al., 2024) as the reinforcement learning framework, and all experiments are conducted on 8 NVIDIA H20 96G GPUs. During the GAO iterative stage, the number of game rounds is set to $T = 4$. In each round the poser samples $M = 4096$ problems. For each generated problem, the solver’s GRPO rollout sampling count is set to $N = 8$. The sampling temperature is set to 1.0, with top-p = 0.7. The thresholds for difficulty-based filtering were set as a pass rate of $P_{low} = 25.0\%$ and $P_{high} = 62.5\%$.

Baselines The baselines consist of basemodels and fine-tuned models.

Base Models are pretrained on large-scale general knowledge corpora and serve as foundational pretrained backbones for subsequent fine-tuning on mathematical tasks. The base models include Qwen3-8B-Base, Qwen3-14B-Base (Yang et al., 2025) and Qwen2.5-7B-Base, Qwen2.5-14B-Base (Team, 2024).

Fine-tuned Models are adapted from base models through fine-tuning on smaller, domain-specific mathematical datasets. The fine-tuned models include Qwen2.5-Math-7B-Instruct (Team, 2024), SVS Liang et al. (2025), PromptCot (Yao et al., 2024), OpenMath-Nemotron-14B (Moshkov et al., 2025), lightR1 (Wen et al., 2025) and Critique-GRPO (Zhang et al., 2025). In addition, we also evaluate models that are not specifically fine-tuned on mathematical datasets but demonstrate strong general reasoning ability, such as DeepSeek-R1-Distill-Qwen-14B (DeepSeek-AI et al., 2025), MEGA-SCIENCE (Fan et al., 2025), AZR (Zhao et al., 2025) and Qwen3-8B-Instruct (Yang et al., 2025),

Evaluation We evaluate the performance of our model on authoritative mathematics benchmark datasets, including the relatively easy GSM8K (Cobbe et al., 2021), the medium-difficulty MATH-500 (Lightman et al., 2023), and higher-difficulty datasets AIME24, AIME25, and OlymMATH (Sun et al., 2025). To ensure fairness, all model inferences are conducted using the VLLM (Kwon et al., 2023) framework. The inference settings use a temperature of 0.6 and a maximum output length of 65,536 tokens. To reduce the effect of measurement variance, all reported results are averaged over eight runs.

4.2 MAIN RESULTS

The performance of **GAO** on mathematical reasoning benchmarks is summarized in Table 1. GAO achieves strong results on benchmarks of varying difficulty, reaching a pass@1 accuracy of **94.6%** on GSM8K and **94.3%** on MATH500, outperforming baseline models. Moreover, GAO demonstrates substantial improvements on high-difficulty benchmarks, achieving **82.1%** on AIME 2024, **67.5%** on AIME 2025, and **57.2%** on OlymMATH, establishing new state-of-the-art results. This pronounced gain is a direct consequence of GAO’s iterative game-theoretic training: the poser dynamically generates increasingly challenging problems tailored to the solver’s current capabilities, effectively pushing the solver to extend its reasoning depth. As a result, GAO performs well on easier tasks and shows clear improvements on more challenging mathematical problems, illustrating the benefits of adversarial, difficulty-adaptive training.

To ensure a rigorous and objective evaluation, we compared GAO with a set of competitive open-source mathematical reasoning models. As shown in Table 1, GAO performs prominently among 7–8B scale models. On more challenging high-difficulty benchmarks such as AIME24, AIME25, and OlymMATH, GAO demonstrates clear advantages. Notably, compared with other GAN-inspired adversarial training methods like SvS and AZR, GAO can significantly raise the model’s capability ceiling, reflecting the superiority of training the poser and solver separately: it avoids the training instability caused by multiple objectives acting on a single model. Compared with the second-best model of similar size, Qwen3-8B, GAO achieves a notable improvement of 6.0% on AIME24 and 7.6% on OlymMATH, highlighting the effectiveness of our method in handling complex mathematical reasoning tasks. We further compared GAO with larger 14B models. GAO outperforms OpenMath-Nemotron-14B, which is carefully optimized on domain-specific mathematical data, across all benchmarks. In particular, GAO surpasses Qwen3-14B on high-difficulty benchmarks, further demonstrating its robustness in challenging scenarios. These results indicate that the iter-

Models	Backbone	GSM8K	MATH500	AIME24	AIME25	OlymMATH	AVG
Base Models							
Qwen2.5-7B-Base	-	82.0	53.4	12.1	0.0	4.5	30.4
Qwen2.5-14B-Base	-	86.5	55.6	13.3	3.3	3.3	32.4
Qwen3-8B-Base	-	57.6	52.8	15.0	7.5	4.3	27.4
Qwen3-14B-Base	-	87.5	71.4	21.3	10.8	7.2	39.6
14B Fine-tuned Models							
DS-Distill-14B	Qwen2.5	88.7	91.0	65.0	50.0	37.5	66.4
OpenMath-Nemotron-14B	Qwen2.5	93.7	94.5	65.8	49.2	38.1	68.3
Light-R1-14B	Qwen2.5	<u>95.8</u>	<u>94.7</u>	<u>75.0</u>	<u>53.3</u>	<u>40.9</u>	<u>71.9</u>
Qwen3-14B	Qwen3	96.0	96.2	81.2	62.5	55.9	78.4
7-8B Fine-tuned Models							
Llama3.1-8B-Instruct	Llama3.1	81.3	44.0	15.8	0.8	2.5	28.8
SvS-8B	Llama3.1	90.3	62.2	-	-	26.4	59.6
Qwen2.5-Math-7B-Instruct	Qwen2.5	<u>94.4</u>	83.8	14.2	10.0	7.0	41.9
AZR-7B	Qwen2.5	-	72.6	20.0	10.0	38.2	35.2
Light-R1-7B	Qwen2.5	86.5	91.6	59.7	45.3	28.1	62.2
PromptCot-7B	Qwen2.5	92.8	<u>93.7</u>	58.7	49.2	20.5	63.0
MEGASCIENCE-8B	Qwen3	92.7	85.9	32.5	25.8	10.8	49.5
Critique-GRPO-8B	Qwen3	93.1	93.4	68.3	49.2	35.6	67.9
Qwen3-8B	Qwen3	93.9	91.37	<u>76.1</u>	<u>65.6</u>	<u>49.6</u>	<u>75.1</u>
GAO(Ours)	Qwen3	94.6	94.3	82.1	67.5	57.2	79.1

Table 1: Compare the pass@1(%) accuracy of competitive models across various mathematical benchmarks. Highlight boldface values to indicate the best performance within models of the same size, and use underlined values to denote the second-best performance at that size.

Dataset	DeepSeek-Distill-7B	
	Accuracy(↓)	AVG. Reasoning Tokens(↑)
AIME24	55.5	4583
LightR1	49.7	6572
PromptCot	48.9	6305
GAO	46.6	7172

Table 2: Problem difficulty comparison across datasets, measured by reasoning trajectory length and pass rate with DeepSeek-R1-Distill-Qwen-7B.

ative adversarial interplay between the poser and solver enables GAO to overcome the limitations of model scale, substantially enhancing the solver’s mathematical reasoning ability to reach or even exceed the level of larger models.

4.3 ANALYSIS

Difficulty Analysis To measure the difficulty gap between problems generated by Poser and those from other datasets, we randomly sampled up to 300 problems from each dataset and evaluated them using the DeepSeek-Distill-7B model. We quantify problem difficulty using model accuracy, with lower accuracy indicating harder problems, and the average length of reasoning trajectories, with longer trajectories indicating more complex reasoning. As shown in Table 2, GAO generates the most challenging problems: it has the lowest model accuracy (46.6%) and the longest average reasoning trajectories (7172 tokens). By comparison, AIME24 problems are easier, with 55.5% accuracy and 4583 reasoning tokens, while LightR1 and PromptCot lie in between. This result demonstrates GAO’s effectiveness in the adversarial iterative process, producing a greater number of high-difficulty, reasoning-intensive problems.

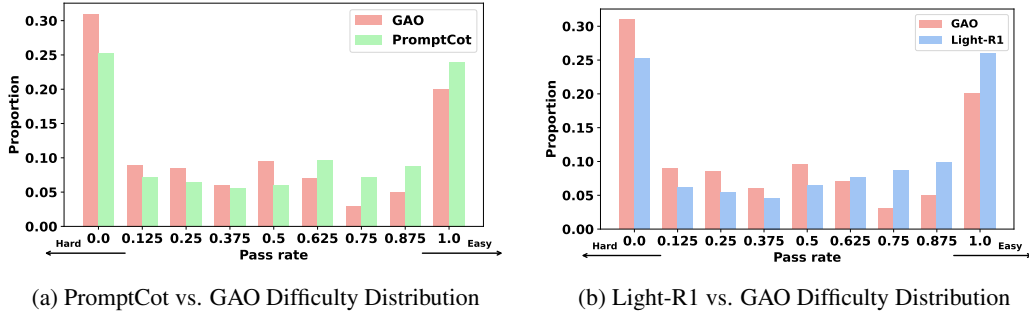


Figure 3: Pass rate distribution of generated problems. GAO produces more difficult and high-value problems, providing richer training signals for solver fine-tuning.

In addition, to investigate whether the generated problems are suitable for training the solver, we employ the model trained from GAO to evaluate both the problems generated by PromptCot, Light-R1 and those generated by GAO. We randomly sample 300 problems from each source and perform 8 inference runs for each problem, then analyze the distribution of samples across different pass rate intervals. Intuitively, if a larger proportion of problems fall into lower pass rate intervals, this indicates higher difficulty. If a larger proportion lies between 25% and 62.5%, it suggests that the generated problems have moderate difficulty relative to the current reasoning model, making them more suitable for reinforcement learning training. Figure 3(a) and Figure 3(b) compare the difficulty distributions of GAO-generated problems with those from the PromptCot and Light-R1 datasets. GAO-generated problems are more heavily concentrated near a pass rate of 0, highlighting their increased difficulty. Moreover, a notable fraction of problems falls within the 25%–62.5% pass rate range, suggesting that GAO’s poser generates a larger proportion of high-value problems. This distribution offers richer training signals, thereby providing more effective guidance for reinforcement learning optimization.

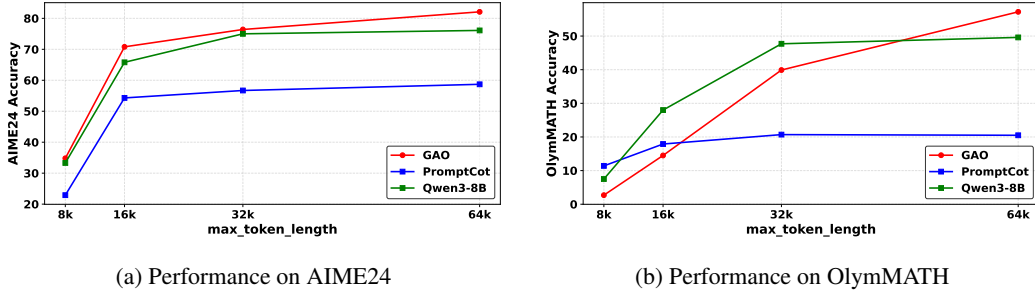


Figure 4: Effect of token budget on GAO’s chain-of-thought reasoning and problem-solving performance on AIME24 and OlymMATH.

Reasoning Ability Analysis Figure 4 illustrates that on the AIME24 and OlymMATH benchmark. As the token budget increases, our GAO model is able to more effectively leverage chain-of-thought (CoT) reasoning, thereby achieving stronger problem-solving performance. Recent works in mathematical reasoning tasks, such as PromptCoT, DeepSeek-R1, and Qwen3, have highlighted the critical role of extending the chain-of-thought in solving challenging problems, and have emphasized the importance of CoT length and quality during dataset construction to enhance reasoning capabilities. From the results in Figure 4(b), we observe that GAO is capable of autonomously extending its CoT length on harder problems, thereby lengthening its reasoning process. Increasing the token budget not only prevents premature truncation due to budget constraints but also ensures the completeness of the reasoning chain, allowing the model’s reasoning capacity to be fully utilized.

In Figure 4(b), we observe pronounced differences in scaling behavior on high-difficulty problems. While Qwen3-8B and PromptCot begin to plateau as the token budget nears 64k, GAO shows a much milder tendency toward saturation, with performance still improving in the large-context regime.

This pattern shows that GAO effectively uses extra tokens to form longer, coherent reasoning chains, yielding more accurate solutions. Unlike baselines, it converts extended context into consistent performance gains, highlighting its ability to exploit long-context information for reliable problem solving.

Iterative Game Dynamics To evaluate the effectiveness of iterative adversarial training, we examine solver performance across successive rounds of poser–solver interactions. We conduct four rounds so that the poser can progressively generate more challenging problems, while the solver has ample opportunity to refine its reasoning strategies. By repeatedly engaging both components in this game-like setup, the poser incrementally escalates problem difficulty, while the solver is continuously driven to refine its reasoning ability.

<i>Turn</i>	AIME24	AIME25	OlymMATH	AVG
0	74.8	60.3	47.5	60.9
1	77.3	62.5	49.3	63.0
2	79.4	64.5	55.3	66.4
3	81.4	66.9	56.7	68.3
4	82.1	67.5	57.2	68.9

Table 3: Effect of game iterations on solver performance. Accuracy improves consistently with more iterations, especially on harder benchmarks.

Table 3 reports solver accuracy on AIME24, AIME25, and OlymMATH for different numbers of game iterations. *Turn* refers to the number of adversarial rounds between the Poser and the Solver. We observe a clear upward trend: performance increases steadily from 60.9% at initialization to 68.9% after four iterations. Notably, the largest relative gain appears on the more difficult OlymMATH (+9.7 points), suggesting that iterative interactions help with harder problems.

Although accuracy improves consistently, the performance gains begin to taper after the third turn, indicating diminishing returns with further iterations. Based on this observation, we limit the number of rounds to four to balance improvements with computational cost. These results highlight that iterative poser–solver dynamics not only produce steady gains across benchmarks but also offer the most pronounced benefits for tasks requiring deeper reasoning, thereby enhancing both robustness and generalization of the solver.

5 CONCLUSION, LIMITATION AND FUTURE WORKS

Current mathematical reasoning solvers often hit a performance ceiling during iterative training, primarily due to training data that is either insufficiently challenging or poorly aligned with the solver’s capabilities. To address this, we introduce GAO, a novel game-based framework where a solver and a poser engage in dynamic, competitive interactions. Through this interplay, the poser incrementally generates problems that expose the solver’s weaknesses, while the solver continuously adapts its reasoning strategies, producing targeted reinforcement learning data of appropriate difficulty.

Our experiments show that GAO achieves state-of-the-art performance across multiple benchmarks, with particularly notable improvements on more challenging problems. However, analysis indicates that GAO’s problem-solving performance relies on longer reasoning chains, which leads to increased resource requirements during inference.

We hope future research can build on GAO by incorporating more efficient reasoning strategies, such as adaptive chain-of-thought pruning or resource-aware inference techniques, enabling the model to maintain high problem-solving performance while keeping computational requirements manageable and scalable for larger benchmarks.

ETHICS STATEMENT

During the data generation process of GAO, although a portion of the generated data was manually reviewed by the authors and the model is intended to be open-sourced, the dataset may still contain content that could be considered harmful or inconsistent with human ethical standards. We encourage users to exercise caution and apply appropriate filtering when utilizing this dataset in future research.

REPRODUCIBILITY STATEMENT

To promote reproducibility and facilitate the review process, we provide data processing scripts at the anonymous link https://anonymous.4open.science/r/anonymous_test-0D24/ and plan to publicly release the trained models for testing in the future. Additionally, we provide a detailed description of the dataset construction process and experimental settings to further enhance reproducibility. This ensures that other researchers can validate our findings and build upon our work, fostering collaboration and advancing progress in the field.

REFERENCES

- Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.
- Essential AI, :, Darsh J Shah, Peter Rushton, Somanshu Singla, Mohit Parmar, Kurt Smith, Yash Vanjani, Ashish Vaswani, Adarsh Chaluvaraju, Andrew Hojel, Andrew Ma, Anil Thomas, Anthony Polloreno, Ashish Tanwer, Burhan Drak Sibai, Divya S Mansingka, Divya Shivaprasad, Ishaan Shah, Karl Stratos, Khoi Nguyen, Michael Callahan, Michael Pust, Mrinal Iyer, Philip Monk, Platon Mazarakis, Ritvik Kapila, Saurabh Srivastava, and Tim Romanski. Rethinking reflection in pre-training, 2025. URL <https://arxiv.org/abs/2504.04022>.
- Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805, 2023. doi: 10.48550/ARXIV.2312.11805. URL <https://doi.org/10.48550/arXiv.2312.11805>.
- Chenjia Bai, Lingxiao Wang, Lei Han, Animesh Garg, Jianye Hao, Peng Liu, and Zhaoran Wang. Dynamic bottleneck for robust self-supervised exploration. *Advances in Neural Information Processing Systems*, 34:17007–17020, 2021.
- Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, et al. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*, 2024.
- Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. *CoRR*, abs/2501.17161, 2025. doi: 10.48550/ARXIV.2501.17161. URL <https://doi.org/10.48550/arXiv.2501.17161>.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948. URL <https://doi.org/10.48550/arXiv.2501.12948>.
- Run-Ze Fan, Zengzhi Wang, and Pengfei Liu. Megascience: Pushing the frontiers of post-training datasets for science reasoning. *arXiv preprint arXiv:2507.16812*, 2025. URL <https://arxiv.org/abs/2507.16812>.
- Huawen Feng, Pu Zhao, Qingfeng Sun, Can Xu, Fangkai Yang, Lu Wang, Qianli Ma, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Qi Zhang. Warriorcoder: Learning from expert battles to augment code large language models. In *Proceedings of the 65th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, Vienna, Austria, July 2025. Association for Computational Linguistics.
- Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars, 2025. URL <https://arxiv.org/abs/2503.01307>.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
- Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5767–5777, 2017. URL <https://proceedings.neurips.cc/paper/2017/hash/892c3b1c6dccc52936e27cbd0ff683d6-Abstract.html>.
- Joshua Hare. Dealing with sparse rewards in reinforcement learning. *arXiv preprint arXiv:1910.09281*, 2019.
- Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.
- Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 1049–1065. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-ACL.67. URL <https://doi.org/10.18653/v1/2023.findings-acl.67>.
- Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model

- serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.
- Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nanning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. Common 7b language models already possess strong math capabilities. *CoRR*, abs/2403.04706, 2024. doi: 10.48550/ARXIV.2403.04706. URL <https://doi.org/10.48550/arXiv.2403.04706>.
- Xiao Liang, Zhongzhi Li, Yeyun Gong, Yelong Shen, Ying Nian Wu, Zhijiang Guo, and Weizhu Chen. Beyond pass@1: Self-play with variational problem synthesis sustains rlvr, 2025. URL <https://arxiv.org/abs/2508.14029>.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding rl-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.
- Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jian-Guang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL <https://openreview.net/forum?id=mMPMHWOdOy>.
- Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset. *arXiv preprint arXiv:2504.16891*, 2025.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel

- Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Sel-sam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Pre-ston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vi-jayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wo-jciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feed-back. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Informa-tion Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/blfede53be364a73914f58805a001731-Abstract-Conference.html.
- Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforce-ment learning. In *International conference on machine learning*, pp. 2817–2826. PMLR, 2017.
- Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. Reasoning with language model prompting: A survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5368–5393, 2023.
- Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.
- Haoxiang Sun, Yingqian Min, Zhipeng Chen, Wayne Xin Zhao, Zheng Liu, Zhongyuan Wang, Lei Fang, and Ji-Rong Wen. Challenging the boundaries of reasoning: An olympiad-level math benchmark for large language models, 2025. URL <https://arxiv.org/abs/2503.21380>.
- Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *CoRR*, abs/2211.09085, 2022. doi: 10.48550/ARXIV.2211.09085. URL <https://doi.org/10.48550/arXiv.2211.09085>.
- Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping

- Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, and Zonghan Yang. Kimi k1.5: Scaling reinforcement learning with llms. *CoRR*, abs/2501.12599, 2025. doi: 10.48550/ARXIV.2501.12599. URL <https://doi.org/10.48550/arXiv.2501.12599>.
- Qwen Team. Qwen2.5: A party of foundation models, September 2024. URL <https://qwenlm.github.io/blog/qwen2.5/>.
- Zengzhi Wang, Rui Xia, and Pengfei Liu. Generative AI for math: Part I - mathpile: A billion-token-scale pretraining corpus for math. *CoRR*, abs/2312.17120, 2023. doi: 10.48550/ARXIV.2312.17120. URL <https://doi.org/10.48550/arXiv.2312.17120>.
- Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. Light-rl: Curriculum sft, DPO and RL for long COT from scratch and beyond. *CoRR*, abs/2503.10460, 2025. doi: 10.48550/ARXIV.2503.10460. URL <https://doi.org/10.48550/arXiv.2503.10460>.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- Junyi Yao, Yijiang Liu, Zhen Dong, Mingfei Guo, Helan Hu, Kurt Keutzer, Li Du, Daquan Zhou, and Shanghang Zhang. Promptcot: Align prompt distribution via adapted chain-of-thought. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7027–7037, 2024.
- Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL <https://openreview.net/forum?id=N8N0hgNDRt>.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025a.
- Qiyang 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, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: an open-source LLM reinforcement learning system at scale. *CoRR*, abs/2503.14476, 2025b. doi: 10.48550/ARXIV.2503.14476. URL <https://doi.org/10.48550/arXiv.2503.14476>.
- Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. Scaling relationship on learning mathematical reasoning with large language models. *CoRR*, abs/2308.01825, 2023. doi: 10.48550/ARXIV.2308.01825. URL <https://doi.org/10.48550/arXiv.2308.01825>.
- 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? *arXiv preprint arXiv:2504.13837*, 2025.
- Xiaoying Zhang, Hao Sun, Yipeng Zhang, Kaituo Feng, Chaochao Lu, Chao Yang, and Helen Meng. Critique-grpo: Advancing llm reasoning with natural language and numerical feedback. *arXiv preprint arXiv:2506.03106*, 2025.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data, 2025. URL <https://arxiv.org/abs/2505.03335>.

A CASE STUDY

A.1 PROMPTS FOR POSER INITIALIZATION

As an experienced education expert, you need to analyze the most core knowledge point from a math problem and act as a question setter to think about how to build the question step by step from this knowledge point.
EXAMPLE
Question: {example_question}
Model Answer: {example_answer}
NOW
Question: {problem}
Model Answer: <concept>...</concept>
<think>...</think>
(If the question is in English, then your answer should also be in English.)

Figure 5: Effect of token budget on GAO’s chain-of-thought reasoning and problem-solving performance on OlymMATH.

As shown in Figure 5, we utilize the prompt to guide Qwen2.5-72B-Instruct in producing both the underlying problem concepts and their construction process. The construction process is then concatenated with the original problem through simple rules, yielding the answer component used for initializing the poser’s SFT training. For clarity and brevity, the manually constructed example question and example answer are not included in the paper, but they were used internally to guide the model during experiments.

A.2 EXAMPLE OF POSER INPUT AND OUTPUT

Here we provide an example of a poser-generated problem, as illustrated in Table 4. The poser requires only a fixed prefix and a defined knowledge scope to generate a problem. Given this prompt, the poser produces a chain of thought that simulates a mathematical expert incrementally constructing a coherent and appropriately challenging question. As shown in Table 5, the problem-solving reasoning is enclosed between *< think >* and *< /think >*, while the final generated problem appears after the chain of thought, facilitating straightforward extraction of the problem itself.

Poser Prompt Example

As an experienced mathematics education expert, you are now tasked with designing a challenging math problem from scratch, based on a broad area of knowledge. The goal is to thoroughly assess students’ abilities. During the problem construction process, you must ensure that the problem is both solvable and sufficiently difficult. The knowledge point is as follows: L’Hôpital’s Rule and Taylor Series Expansion in the evaluation of a limit

Table 4: Input prompt for the poser.

Poser Generated Problem Example

< think >When designing a question that involves the application of L'Hôpital's Rule and Taylor Series Expansion in the evaluation of a limit, as an educational expert, I focus on several key aspects to ensure the problem is both educational and challenging:

1. Mathematical Background and Concept Introduction

Context Selection: The problem is set in the context of limit evaluation, a fundamental topic in calculus. The limit involves an indeterminate form $\frac{0}{0}$ as $x \rightarrow 0$, which is a classic scenario where L'Hôpital's Rule can be applied. Incorporating Trigonometric and Logarithmic Functions: The problem includes $\sin x$ and $\ln(1+x)$, which are common in calculus and provide a good opportunity to explore the behavior of these functions near zero.

2. Parameter Setting and Function Selection

Numerator and Denominator: The numerator is $(\sin x - \ln(1+x))^2$, and the denominator is $x^3 \cdot \tan(2x^3)$. Both the numerator and the denominator approach zero as $x \rightarrow 0$, making the limit an indeterminate form $\frac{0}{0}$. Complexity of the Expression: The squared term in the numerator adds a layer of complexity, requiring the student to apply L'Hôpital's Rule more than once or to use series expansions to simplify the expression.

3. Teaching Goals and Skill Development

L'Hôpital's Rule Application: The problem tests the student's ability to apply L'Hôpital's Rule correctly, especially in cases where the rule needs to be applied multiple times. Taylor Series Expansion: The problem also encourages the use of Taylor series expansions, which can simplify the limit evaluation process by approximating the functions involved. Algebraic Manipulation: Students need to manipulate the expression to simplify it, which involves combining and rearranging terms.

4. Step-by-Step Problem Solving

Identify the Indeterminate Form: Recognize that both the numerator and the denominator approach zero as $x \rightarrow 0$. Apply L'Hôpital's Rule: Differentiate the numerator and the denominator with respect to x and simplify the resulting expression. Check for Further Indeterminate Form: After applying L'Hôpital's Rule once, if the limit is still in an indeterminate form, apply the rule again. Use Series Expansions: If the limit is still not straightforward, use the Taylor series expansions of $\sin x$ and $\ln(1+x)$ to approximate the numerator and the denominator. Simplify and Evaluate: Combine the approximations and simplify the expression to find the limit.

5. Educational Objectives and Skill Enhancement

Conceptual Understanding: Students will deepen their understanding of L'Hôpital's Rule and the utility of Taylor series expansions in limit evaluation. Problem-Solving Skills: The problem encourages students to think critically about how to handle complex expressions and apply multiple techniques to reach a solution. Mathematical Rigor: The problem promotes the development of rigorous mathematical reasoning and the ability to handle multiple steps in a problem-solving process.

6. Logical Flow of the Question

Set the Context: Introduce the limit $\lim_{x \rightarrow 0} \frac{(\sin x - \ln(1+x))^2}{x^3 \tan(2x^3)}$. Identify the Indeterminate Form: Show that both the numerator and the denominator approach zero as $x \rightarrow 0$.

Apply L'Hôpital's Rule: Differentiate the numerator and the denominator and simplify the expression. Check for Further Indeterminate Form: If necessary, apply L'Hôpital's Rule again. Use Series Expansions: If L'Hôpital's Rule is not sufficient, use the Taylor series expansions of $\sin x$ and $\ln(1+x)$ to approximate the functions. Simplify and Evaluate: Combine the approximations and simplify the expression to find the limit.

By following these steps, the problem aims to challenge students to apply advanced calculus techniques and enhance their problem-solving skills in a structured and educational manner.< /think >

Compute the following limit: $\lim_{x \rightarrow 0} \frac{(\sin x - \ln(1+x))^2}{x^3 \tan(2x^3)}$

Table 5: Example of a poser-generated problem.

B ADDITIONAL STATEMENTS

B.1 THE USE OF LARGE LANGUAGE MODELS (LLMs)

In our paper, LLMs were used during the iterative stage of GAO data synthesis, with specific examples provided in Appendix A. Aside from this, LLMs were only employed for polishing the writing of the paper.