HINT-GUIDED DIVERSIFIED POLICY OPTIMIZATION FOR LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent developments in Large Language Models (LLMs) have showcased impressive reasoning capabilities and a promising enhancement strategy is Reinforcement Learning with Verifiable Rewards (RLVR). However, existing reward mechanisms are constrained to the outcome-level correctness and lack explicit signals to guide the model to consider diverse solutions. In contrast, human problem-solving usually involves considering several potential solutions and choosing the most reliable one, a cognitive process that current RLVR frameworks do not explicitly incentivize. Inspired by this cognitive process, we propose Hint-Guided Diversified Policy Optimization (HDPO), allowing the model to first list all potential candidate solution outlines as hints and then select the most reliable one for further reasoning. HDPO comprises two stages of **Cold Start for Structured Reasoning** and **Hint-Guided Diversified Reinforcement Learning** to incentivize the model to generate diverse and reliable solutions following the "propose-select-think" trajectory. Experimental results show that HDPO effectively boosts LLM reasoning and enhances the diversity of candidate solutions as well as the LLM's ability to identify reliable solutions across mathematical and knowledge-intensive tasks.

1 INTRODUCTION

Recent research has advanced the reasoning capabilities of Large Language Models (LLMs), including GPT-5 (Singh et al., 2025), DeepSeek-V4 (Xu et al., 2026), and Kimi K3 (Team et al., 2026), particularly in complex mathematical problem solving. Reinforcement Learning with Verifiable Reward (RLVR) (Liu et al., 2025b; Yan et al., 2025; Zhang et al., 2025a) has become a core training paradigm for these advances. Within this framework, policy optimization algorithms are widely adopted to fine-tune LLMs, with Group Relative Policy Optimization (GRPO) (Shao et al., 2024) serving as a prominent example of policy optimization for complex reasoning tasks.

To ensure the credibility of complex reasoning, outcome-level rewards (Cobbe et al., 2021; Shao et al., 2024) are employed in RLVR to assess the correctness of the entire reasoning trajectory. However, this approach does not take into account the rationality of the process, and its sparse rewards may lead to homogenization of rewards, failing to provide comprehensive supervision. Therefore, recent studies (Zhang et al., 2025b;a; Luo et al., 2025b) enhance reasoning with more detailed feedback, including natural-language feedback and process supervision.

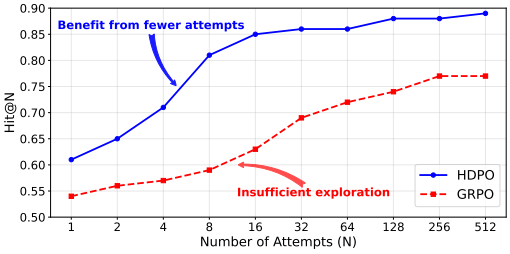


Figure 1: Hit@N for HDPO and GRPO on Olympiad-Bench at varying attempt counts.

We noticed that although the above methods have achieved notable success, they provide rewards based exclusively on correctness and thus fail to encourage the exploration of diverse solutions. Consequently, if the model gets trapped in a wrong solution, it may struggle to recover and identify the correct answer, reducing the fault tolerance of reasoning. As shown in Figure 1, we evaluate the Hit@N accuracy of Qwen3-4B across varying numbers of sampling attempts. The results reveal that GRPO exhibits markedly lower performance when the number of attempts is limited, whereas our proposed HDPO consistently achieves superior accuracy even with fewer attempts. This sug-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

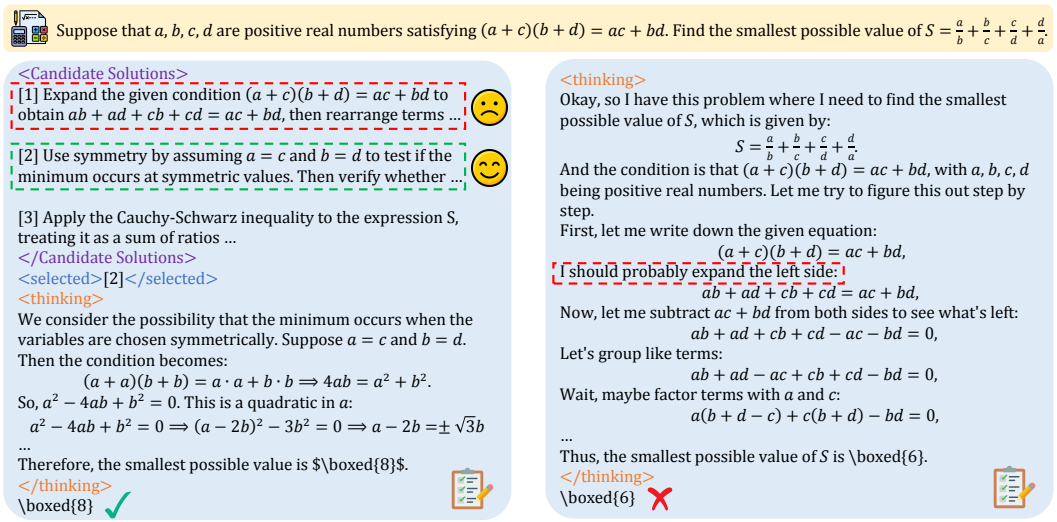


Figure 2: The comparison of HDPO's "propose-select-think" reasoning process (**left**) and the traditional reasoning process (**right**). More case studies are provided in Appendix K.

gests that the GRPO-trained model has insufficiently explored the space of diverse reasoning paths, which hinders its ability to generate correct answers within a limited number of trials. Ideally, the model should be able to explore a broader range of solutions and choose the most reliable solutions for further refinement. The key challenge lies in designing mechanisms that both incentivize the generation of diverse solutions and effectively assess their reliability.

To address this issue, we propose Hint-Guided Diversified Policy Optimization (HDPO), a novel approach to enhance the diversity and reliability of LLM reasoning. HDPO is built upon the "propose-select-think" reasoning trajectory, effectively stimulating the model to reason about diverse potential solutions. Specifically, our training methodology instills this capability in two stages: we first employ supervised fine-tuning (SFT) in the cold start stage to establish the model's proficiency in following the structured reasoning trajectory, followed by Reinforcement Learning (RL) to cultivate the policy model capable of generating diverse and reliable solutions. In the cold start phase, we employ an advanced LLM to construct such structured reasoning trajectories and apply filtering criteria based on correctness and reliability to ensure data quality. In the RL phase, we introduce diversity scheduling strategy and reliability reward to guide the model to explore all potential solutions.

As illustrated in Figure 2, the "propose-select-think" framework employed by HDPO offers a significant advantage over the traditional approach. While the conventional paradigm extends the given condition in a brute-force manner and arrives at an incorrect answer, HDPO promotes broader exploration of the solution space by generating multiple candidate solutions prior to engaging in detailed reasoning. This preliminary exploration enables the model to identify underlying symmetries, a critical insight that contributes to the correct answer.

Crucially, HDPO advances existing paradigms through **policy internalization** and **joint optimization**. Unlike inference-time search methods (Wang et al., 2023; Yao et al., 2023) that incur high latency via repeated sampling, HDPO internalizes the explore-then-exploit mechanism into the policy parameters via RLVR, enabling candidate proposal, selection, and reasoning within a single generated trajectory. Furthermore, departing from accuracy-centric RLVR (Zhang et al., 2025a; Yan et al., 2025), it directly mitigates solution homogenization and unreliable selection by jointly optimizing a scheduled diversity reward and a confidence-based reliability reward. To summarize, our contributions are as follows:

- We introduce HDPO to enhance LLM reasoning, a novel framework based on the "propose-select-think" structured reasoning process.
- The proposed diversity scheduling strategy and reliability reward enable LLMs to actively explore diverse and reliable solutions.
- Our extensive experimental results demonstrate that HDPO greatly boosts the model's reasoning ability.

2 Related Work

Chain-of-Thought for LLM Reasoning. Chain-of-thought (Wei et al., 2022) reasoning has shown promising progress in recent years. Building on this idea, tree-of-thought (ToT) (Yao et al., 2023) generalizes reasoning into structured sequences of thoughts and determines the next action through self-evaluation. Zhou et al. (2023) propose least-to-most prompting by decomposing problems, enabling language models to solve problems that are more difficult than those in the prompts. Graph of Thoughts (GoT) (Besta et al., 2024) models LLM reasoning as an arbitrary graph, where thoughts are represented as vertices and dependencies between thoughts are represented as edges. Unlike ToT and GoT, which rely on repeated sampling and heuristic scoring at inference, HDPO uses reward-shaped RL to train the model to propose diverse candidate hints and select a reliable one to guide subsequent reasoning.

Reinforcement Learning for LLM Reasoning. Recent advances in reinforcement learning (Shao et al., 2024; Yue et al., 2025; Liu et al., 2025a) have substantially improved the reasoning capabilities of LLMs, shifting from surface-level response generation to sophisticated problem-solving. Ouyang et al. (2022) utilize preference data to train reward models and apply Proximal Policy Optimization (Schulman et al., 2017) to perform reinforcement learning on LLMs for alignment with human preferences. Subsequently, numerous methods have employed reinforcement learning to align LLMs with human preferences (Lee et al., 2024; Ryu et al., 2024; Xie et al., 2024). DeepSeek-Math (Shao et al., 2024) proposes the GRPO algorithm, which simplifies the training process of reinforcement learning and significantly enhances the performance of LLMs in the mathematical domain through RL. Building on this foundation, Dr.GRPO (Liu et al., 2025b) proposes an unbiased optimization method that improves token efficiency while maintaining reasoning performance. Aggarwal & Welleck (2025) use length constraints to ensure that the reasoning language model adheres to the specified length by the user. To ensure that sequence-level rewarding and optimization are aligned, Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025) defines importance ratios based on sequence likelihood.

3 Methods

3.1 Preliminary

We utilize GRPO (Shao et al. 2024) to optimize the policy model π_θ. Given a prompt p, the old policy $\pi_{\theta_{\text{old}}}$ generates G responses $\{x_1, \ldots, x_G\}$ with rewards $\{r_1, \ldots, r_G\}$. The response-level advantages are computed through group normalization:

$$\hat{A}_i = \frac{r_i - \text{mean}(r_1, \ldots, r_G)}{\text{std}(r_1, \ldots, r_G) + \epsilon_{\text{norm}}}, \tag{1}$$

where ϵ_{norm} is a small constant for numerical stability.

The policy is then optimized using the token-level clipped GRPO objective with KL regularization:

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\Bigg[\min\Big(\rho_{i,t}(\theta)\hat{A}_i, \\ \text{clip}\left(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_i\Big) - \beta D_{\text{KL}}\Big(\pi_\theta \,\Big\|\, \pi_{\text{ref}}\Big)\Bigg], \tag{2}$$

where T_i is the length of response x_i, and $\rho_{i,t}(\theta) = \frac{\pi_\theta(x_{i,t}|p, x_{i,<t})}{\pi_{\theta_{\text{old}}}(x_{i,t}|p, x_{i,<t})}$ is the token-level importance ratio. All tokens in x_i share the response-level advantage $\hat{A}_i$. The hyperparameters ϵ and β control clipping and KL regularization, respectively. GRPO improves LLM reasoning by rewarding positive relative advantages while constraining policy drift.

3.2 Overview

As shown in Figure 3, HDPO establishes a two-stage optimization policy. To enable the model to first propose candidate solutions and then select the most reliable solution for thinking, we perform

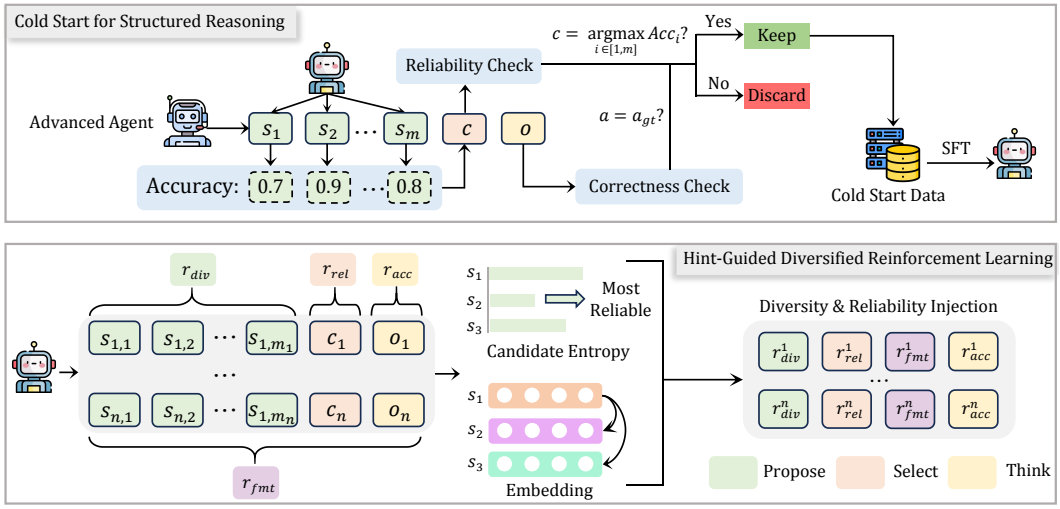


Figure 3: Overview of Hint-Guided Diversified Policy Optimization, comprising two stages: (1) **Cold Start for Structured Reasoning**, which constructs correctness- and reliability-filtered "propose-select-think" trajectories via an advanced LLM; and (2) **Hint-Guided Diversified Reinforcement Learning**, which optimizes the policy via diversity scheduling and reliability rewards to foster solution exploration.

Cold Start for Structured Reasoning in the first stage. Specifically, we distill the data of the "propose-select-think" reasoning trajectory from an advanced LLM to perform SFT. To ensure the validity of these trajectories, we filter the distilled data based on both correctness and reliability. In the second stage, we propose **Hint-Guided Diversified Reinforcement Learning** to enhance the quality of model reasoning through the diversity scheduling strategy and the reliability reward during the process of policy optimization.

3.3 Cold Start for Structured Reasoning

Previous reasoning LLMs (Liu et al., 2025b; Yan et al., 2025; Zhang et al., 2025a) typically involve generating a single solution directly from a given problem, a paradigm that lacks an explicit mechanism to encourage exploration of the diverse solution space. Therefore, we seek to teach the model to follow the "propose-select-think" reasoning trajectory. Specifically, the model initially generates an overview comprising multiple candidate solutions, each delineated in several sentences, and then selects the most reliable candidate to serve as a guiding hint for subsequent thinking.

To fulfill the need for the policy model to follow the reasoning pattern of "propose-select-think", we distill such trajectory data from an advanced teacher LLM. Given a problem q, the teacher model is prompted to generate a structured reasoning trajectory that adheres to the "propose-select-think" framework. The complete prompt used to elicit structured reasoning trajectories is provided in Appendix J. Each trajectory is of the form $\tau = \{s_1, s_2, ..., s_m, c, o\}$, where s_i, c, and o respectively represent the i-th proposed candidate solution, the most reliable solution selected, and the thinking process. To ensure the rationality of the generated data, we apply a two-fold filtering criterion: (1) the correctness of the final answer and (2) the reliability of the selected solution.

We extract the final answer a from the thinking process o, and compare it with the ground truth a_{gt}. If there is an error, we discard this piece of data. Otherwise, we proceed to check the reliability of the selected solution. Specifically, we feed the problem and the m candidate solution overviews sequentially into a lightweight model π_s, allowing it to sample L answers for each candidate solution.

We calculate the accuracy Acc_i of the sampled answers $\mathcal{A}_i$ as follows:

$$\mathcal{A}_i = \{\hat{a}_{i,j}\}_{j=1}^{L} \sim \pi_s(q, s_i), \tag{3}$$

$$Acc_i = \frac{\sum_{\hat{a}_{i,j} \in \mathcal{A}_i} \mathbf{1}[\hat{a}_{i,j} = a_{gt}]}{|\mathcal{A}_i|}. \tag{4}$$

Our key insight is that more reliable candidate solutions are also conducive to the model generating accurate answers. If the optimal candidate solution c selected by the model aligns with the candidate solution exhibiting the highest accuracy, i.e., $c = \arg\max_{i\in[1,m]} Acc_i$, then it indicates that the solution selected by the model is reliable, and this piece of data is retained.

After constructing the cold start training data $\mathcal{D}_{cssr} = \{(q_i, \tau_i)\}_{i=1}^{n}$, we optimize the policy model π_θ through supervised fine-tuning:

$$\hat{\theta} = \arg\min_{\theta} -\mathbb{E}_{(q,\tau)\sim\mathcal{D}_{\text{cssr}}} \left[\log \pi_\theta(\tau \mid q)\right], \tag{5}$$

where $\hat{\pi}_\theta$ is the policy model after supervised fine-tuning during the cold start phase.

3.4 Hint-Guided Diversified Reinforcement Learning

Following the cold-start phase, the policy model has acquired the structured reasoning ability of "propose-select-think". To further enhance its ability to navigate a broader solution space and select high-quality candidates, we refine the hints associated with candidate solutions through reinforcement learning. For each problem q, the model $\hat{\pi}_\theta$ samples G reasoning trajectories in each rollout. The form of the i-th reasoning trajectory is $x_i = \{s_{i,1}, s_{i,2}, ..., s_{i,m_i}, c_i, o_i\}$, where m_i represents the number of candidate solutions generated in the i-th reasoning trajectory. For notational convenience, we henceforth omit the subscript corresponding to each trajectory, i.e., each reasoning trajectory is denoted as $x = \{s_1, s_2, ..., s_m, c, o\}$.

Format Reward. To ensure that the model's output follows the predefined "propose-select-think" structure, we define the format reward. We determine (1) whether the model's output contains the tags "`<Candidate Solutions></Candidate Solutions>`", "`<selected></selected>`", and "`<thinking></thinking>`"; (2) whether the number of candidate-solution overviews is between 1 and 5; (3) whether exactly one reliable solution is selected. A reward of $r_{fmt} = 0.1$ is granted for strict adherence to the structure, while any deviation results in a reward of $r_{fmt} = 0$.

Accuracy Reward. To ensure the accuracy of the final answer, we extract the answer a from the model's reasoning trace o, and judge whether it is consistent with the ground truth a_{gt}:

$$r_{acc} = \begin{cases} 1, & \text{if } a = a_{gt}, \\ 0, & \text{if } a \neq a_{gt}. \end{cases} \tag{6}$$

Therefore, the model receives positive feedback when its output matches the ground truth.

Diversity Reward. We aim to encourage the model to explore a broader solution space rather than being confined to a single solution, as this facilitates the selection of more reliable solutions. Therefore, the candidate solutions should be pairwise distinct to ensure the diversity of solutions. To quantify the similarity among candidate solutions, candidate solutions are encoded into embeddings $\mathcal{E} = \{e_1, \ldots, e_m\}$, and similarity is quantified as the average pairwise cosine similarities:

$$\alpha_{sim} = \frac{1}{m(m-1)} \sum_{i\in[1,m]} \sum_{j\in[1,m]\setminus\{i\}} \cos(e_i, e_j). \tag{7}$$

Equation (7) applies for $m \geq 2$. For $m = 1$, pairwise diversity is unobserved, so we use the mean $\bar{d} = |\mathcal{V}|^{-1} \sum_{j\in\mathcal{V}} (1 - \alpha_{sim}^{(j)})$ of correct multi-candidate rollouts $\mathcal{V}$ in the same group as a neutral reference. If $\mathcal{V}$ is empty, the group receives no diversity term. In the early stages of training, the policy model should prioritize solution correctness to establish a solid foundation of reasoning. Once its ability is sufficiently strong, we gradually shift the optimization objective toward promoting solution diversity. To this end, we have designed a diversity scheduling strategy, using sine learning coefficients to gradually increase the proportion of diversity rewards during the training process. Therefore, the diversity reward for the t-th step in the training process is as follows:

$$r_{div}^{(i)} = \mu\, d_i \sin\left(\frac{\pi}{2} \cdot \frac{\max(0, t - t_{wp})}{t_{max} - t_{wp}}\right), \quad d_i = \begin{cases} 1 - \alpha_{sim}^{(i)}, & m_i \geq 2, \\ \bar{d}, & m_i = 1. \end{cases} \tag{8}$$

where μ is the hyperparameter controlling the level of diversity, t_{wp} is the number of warm-up steps before the diversity reward is applied and t_{max} is the maximum number of training steps. Under

this scheduling strategy, the weight coefficient of diversity gradually increases from 0 to μ. As the model's capability improves, we refine its reasoning process to generate diverse candidate solutions while preserving answer correctness.

Reliability Reward. During the reinforcement learning phase, it is essential to maintain diversity among candidate solutions while simultaneously ensuring that the solution selected by the model is the most reliable. However, assessing reliability via the LLM-based sampling method would incur substantial computational overhead. Therefore, we employ the entropy of the candidate solutions generated by the model as a proxy to judge the reliability of the solutions. Candidate solutions with lower entropy reflect higher model confidence and are more likely to lead to the correct answer. The uncertainty of each candidate solution is quantified by computing the mean entropy across its constituent tokens:

$$H_t = -\sum_{v=1}^{|V|} p_v^{(t)} \log p_v^{(t)}, \qquad \bar{H} = \frac{1}{T}\sum_{t=1}^{T} H_t, \tag{9}$$

where $p_v^{(t)}$ is the probability of the v-th token in the vocabulary V at the t-th position, H_t is the entropy corresponding to the t-th token in the candidate solution, and T is the number of tokens in the candidate solution. The candidate solutions s_1 to s_m are sorted in ascending order of entropy to obtain $\mathcal{S} = \{\hat{s}_1, \hat{s}_2, ..., \hat{s}_m\}$, then the reliability reward corresponding to this reasoning trajectory is r_{rel}:

$$r_{rel} = \frac{1}{k}, \ \text{s.t. } s_c = \hat{s}_k, \tag{10}$$

which means that if the most reliable solution c chosen by the model is ranked k-th among all candidate solutions, the reliability reward is $r_{rel} = \frac{1}{k}$. In this manner, the model is incentivized to select candidate solutions that exhibit greater reliability.

By incorporating the aforementioned four rewards, we guide the model to adhere to the "propose-select-think" trajectory, generating diverse and reliable solutions. The overall reward of the i-th sample in the rollout is as follows,

$$r_i = r_{fmt}^{(i)} + r_{acc}^{(i)} + \mathbf{1}[r_{acc}^{(i)} > 0] \cdot r_{div}^{(i)} + \bar{r}_{\text{acc}} \cdot r_{rel}^{(i)}, \tag{11}$$

where $\mathbf{1}[r_{acc}^{(i)} > 0]$ indicates whether the answer corresponding to the i-th sample is correct, and HDPO applies the diversity reward exclusively when the answer generated by the model is correct. To calibrate the reliability reward and mitigate overconfidence, we scale $r_{\text{rel}}^{(i)}$ by the group-level average accuracy $\bar{r}_{\text{acc}} = \frac{1}{G}\sum_{j=1}^{G} r_{\text{acc}}^{(j)}$. Subsequently, the group-relative advantage is computed based on Equation 1, and the model $\hat{\pi}_\theta$ is optimized using the clipped loss in Equation 2.

4 Experiments

4.1 Experimental Setup

Datasets and Benchmarks. In the cold-start phase, DAPO-Math-17k (Yu et al., 2025) was utilized as seed data to generate thinking trajectories. Subsequently, DeepScaleR-Preview-Dataset (Luo et al., 2025a) was employed for the reinforcement learning stage. We evaluated HDPO on nine reasoning benchmarks: AMC 2023 (MAA), AIME 2024 (Maxwell-Jia, 2024), AIME 2025 (math ai, 2025), Math-500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), Olympiad-Bench (He et al., 2024), MMLU-Pro (Wang et al., 2024b), SciBench (Wang et al., 2024a), and GPQA-Diamond (Rein et al., 2023).

Implementation Details. We experimented with using `Qwen3-4B` (Yang et al., 2025), `Qwen2.5-Math-7B` (Yang et al., 2024), and `DeepSeek-R1-Distill-Llama-8B` (Guo et al., 2025) as backbone models. In order to construct cold start data based on seed data, we employ `Qwen3-235B-A22B-Instruct-2507` to generate thought trajectories and another lightweight model `Qwen3-1.7B` to measure the reliability of each candidate solution. More implementation details can be found in Appendix A.

Baselines and Evaluation. We compared HDPO with the standard GRPO trained on DAPO-Math-17k and DeepScaleR-Preview-Dataset. Additionally, we also use the following powerful

Table 1: Performance of HDPO and baselines on nine reasoning benchmarks.

Methods	AMC 23	AIME 24	AIME 25	Math	Minerva	Olympiad	GPQA	MMLU-Pro	SciBench	Avg.
Previous RLVR Methods										
EDGE-GRPO	73.79	16.36	19.37	84.10	49.89	48.79	46.76	54.10	25.79	46.55
Oat	73.59	38.91	16.64	85.34	38.75	50.82	44.43	47.72	26.31	46.95
SimpleRL	74.27	36.43	16.34	85.22	40.43	49.84	47.31	46.48	25.72	46.89
LUFFY	79.68	28.95	26.61	87.99	44.54	56.09	61.51	67.63	33.37	54.04
Critique-GRPO	81.88	44.93	28.44	91.22	51.51	61.46	67.28	74.31	41.38	60.27
Qwen3-4B										
Base Model	59.10	20.36	18.73	78.53	39.82	40.11	46.81	57.95	21.93	42.59
GRPO	73.32	32.08	25.78	87.01	49.17	55.50	58.47	68.57	33.92	53.76
HDPO	**83.26**	**43.56**	**27.92**	**92.47**	**54.92**	**62.37**	**65.49**	**76.12**	**43.16**	**61.03**
Qwen2.5-Math-7B										
Base Model	71.08	31.55	15.04	79.49	30.17	43.73	48.72	36.74	17.70	41.58
GRPO	80.92	40.76	24.55	89.72	44.86	53.68	60.18	53.19	35.54	53.71
HDPO	**85.79**	**48.75**	**31.69**	**93.28**	**53.54**	**64.76**	**69.89**	**65.63**	**44.36**	**61.97**
DS-R1-Llama-8B										
Base Model	63.78	16.43	16.08	77.54	37.67	38.88	41.29	48.57	19.03	39.92
GRPO	76.62	24.55	22.32	85.53	45.14	54.74	58.60	59.53	36.10	51.46
HDPO	**84.17**	**37.76**	**30.75**	**92.74**	**56.68**	**56.19**	**63.41**	**67.90**	**41.02**	**58.96**

RLVR-based baselines: Qwen2.5-Math-7B-Oat-Zero (Liu et al., 2025b), Qwen-2.5-7B-SimpleRL-Zoo (Zeng et al., 2025), LUFFY-Qwen-Math-7B-Zero (Yan et al., 2025), EDGE-GRPO-Qwen-7B (Zhang et al., 2025b) and Critique-GRPO-8B (Zhang et al., 2025a). For each test question, we sample 16 responses and calculate the unbiased pass@3 (Chen et al., 2021) metric for evaluation.

4.2 Main Results

We conducted evaluations on different model scales (4B, 7B, and 8B) and model families (Qwen and Llama). As shown in Table 1, HDPO outperforms the RLVR methods on both mathematical reasoning and general reasoning benchmarks. It suggests that HDPO can be adapted to different LLMs. Additionally, since GRPO was employed for model optimization, we also compared with the original GRPO method. It can be observed that HDPO has a significant advantage over GRPO, with improvements of 7.27 and 8.26 respectively on Qwen3-4B and Qwen2.5-Math-7B, which is due to HDPO's ability to encourage the model to explore more potential solutions, thereby choosing more reliable solutions and leading to accurate answers.

It is worth noting that after optimization by HDPO, the performance of Qwen3-4B exceeds that of Critique-GRPO using Qwen3-8B as backbone. Moreover, HDPO not only improves mathematical reasoning capabilities, but also has a stable enhancement in general reasoning abilities, indicating its ability to generalize to other domains.

4.3 Ablation Study

Taking `Qwen2.5-Math-7B` as an example, we analyze each module of HDPO in Table 2.

Table 2: Ablation results for each component in HDPO.

Methods	AIME 25	Math	Olympiad	GPQA
HDPO	**31.69**	**93.28**	**64.76**	**69.89**
w/o Cold Start	28.63	90.55	61.59	65.74
w/o RL	21.47	85.81	51.23	58.13
w/o Diversity	27.24	90.20	60.16	66.29
w/o Div. Scheduling	30.51	91.77	63.43	68.17
w/o Reliability	28.86	90.38	61.46	66.51
CS-Qwen3-14B	29.76	91.24	62.41	67.70
Random Reliability	27.45	89.58	60.73	65.47
Reverse Reliability	26.93	88.60	60.14	63.82
0/1 Reward	29.87	91.30	63.15	67.19

Cold Start and Reinforcement Learning. We experimented with removing the cold start phase ("w/o Cold Start") and directly performing reinforcement learning. The results indicate that the cold start effectively enhances the model's reasoning ability, attributed to its adaptation of the model to the "propose-select-think" reasoning pattern. Additionally, we used a relatively small LLM `Qwen3-14B` to construct the cold start data ("CS-Qwen3-14B"). Although performance decreased compared to using a stronger model, it still surpassed that of GRPO, indicating that higher-quality cold-start data generated by more capable models leads to better initialization. We also analyzed the scenario of directly removing the entire reinforcement

learning phase ("w/o RL") and only performing cold start. The performance dropped compared to removing cold start phase, highlighting the importance of diversity and reliability.

Diversity. During the reasoning process, we encourage the model to actively explore diverse solution spaces by imposing the diversity reward. As shown in Table 2, the model performance decreases significantly after removing the diversity reward ("w/o Diversity Reward"), underscoring the importance of promoting diversity among candidate solutions to foster more thorough reasoning and improve answer accuracy. Recognizing that the model exhibits limited reasoning capability in the early stages of RL training, HDPO gradually increases the proportion of diversity rewards through a diversity scheduling strategy. We can observe that its contribution to HDPO is positive, indicating that such a scheduling strategy is beneficial for balancing the accuracy and diversity.

Reliability. To enable the model to select reliable solutions, we design a reliability reward that evaluates the quality of the selected solution. Removing this reward degrades performance, as the model loses supervision for the selection process and thus becomes less capable of distinguishing effective solutions from suboptimal candidates.

The reliability reward in HDPO is based on the close relationship between solution reliability and confidence. We further consider two variants. "Random Reliability" randomly assigns a reward from $[1, \frac{1}{2}, ..., \frac{1}{m}]$, where m is the number of candidate solutions. "Reverse Reliability" ranks candidate solutions by entropy in descending order and assigns the reciprocal rank of the selected solution as the reward. Both variants perform worse than HDPO, supporting the effectiveness of favoring high-confidence solutions for reasoning.

To capture different degrees of candidate reliability, we adopt a continuous reward inversely proportional to the rank of the selected solution. In contrast, "0/1 Reward" assigns 1 only when the top-ranked solution is selected and 0 otherwise. Its inferior performance indicates that a discrete reward cannot adequately distinguish solutions with different reliability levels.

4.4 Analysis

Trends in Accuracy and Diversity. To maintain the diversity of candidate solutions generated by the model, we introduced diversity reward to encourage differences between solutions. We compared the trend of diversity of candidate solutions and the accuracy of answers during the training process before and after adding diversity rewards.

As shown in Figure 4, compared to the HDPO with diversity reward (i.e., "HDPO"), the performance of HDPO without adding diversity reward (i.e., "w/o Div") reached saturation prematurely. This phenomenon results from the diversity reward directing the model toward a broader solution space rather than constraining optimization to outcome-level correctness. Furthermore, we examined the effect of ablating the diversity scheduling strategy ("w/o Scheduling"). The absence of the scheduling mechanism compromises training stability, which in turn causes the model to overemphasize diversity. Although accuracy initially peaks, it subsequently exhibits fluctuations, ultimately yielding inferior performance compared to the scheduled variant. Across all four benchmarks, Table 2 confirms that removing the diversity reward has a greater effect on performance than omitting its schedule. The training curves in Figure 4 reinforce this finding: the diversity reward broadens exploration, while its schedule stabilizes optimization.

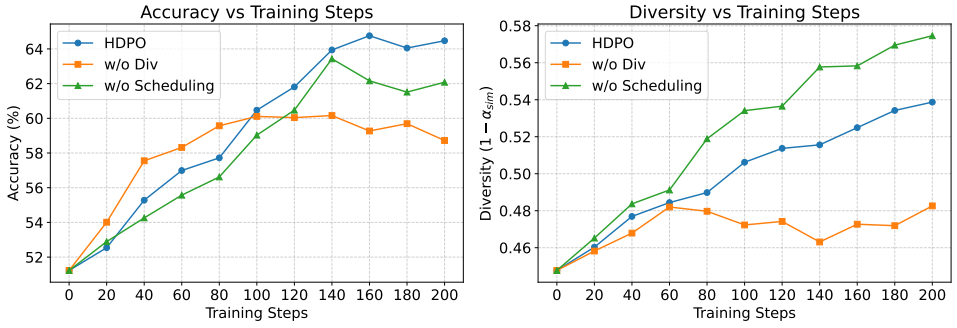


Figure 4: The effect of diversity reward configurations on model performance for Olympiad-Bench.

Confidence–Reliability Correlation. HDPO uses solution confidence as a proxy for reliability. We test this on 10,000 training rollouts by comparing candidate rankings based on confidence and accuracy estimated from LLM samples. Table 3 shows Spearman and Kendall correlations above 0.5 for all three backbones, indicating a consistent positive association.

Table 3: Correlation between confidence and reliability.

Backbone	Spearman	Kendall
Qwen3-4B	0.64	0.61
Qwen2.5-Math-7B	0.67	0.63
DS-R1-Llama-8B	0.62	0.57

We further assess the impact of the reliability reward on selection behavior. Specifically, we extracted the candidate solutions selected by the model, evaluated their reliability based on their performance across multiple independent samples, and assessed whether the model's selections aligned with the most reliable solutions. As shown in Table 4, after applying the reliability reward, the model is able to select reliable solutions more accurately, aligning with the reliability derived from accuracy rate. This indicates that the reliability reward effectively stimulates the model's capacity to identify reliable candidate solutions. The improvement holds for all three backbones and each of the four evaluated reasoning benchmarks.

Table 4: Assessing the impact of the reliability reward on candidate selection accuracy.

Backbones	AIME 25	Math	Olympiad	GPQA
w/o Reliability Reward				
Qwen3-4B	60.63	71.19	64.79	66.19
Qwen2.5-Math-7B	61.46	67.74	62.25	72.13
DS-R1-Llama-8B	57.71	66.54	58.54	63.04
w/ Reliability Reward				
Qwen3-4B	75.64	87.40	80.58	81.50
Qwen2.5-Math-7B	77.29	91.65	82.27	82.17
DS-R1-Llama-8B	71.88	82.51	76.87	75.32

Self-Evolution via Alternating Cold Start and RL. In the cold start phase, we employed an advanced LLM to construct structured reasoning trajectories following the "propose-select-think" framework. To alleviate the dependence on the teacher model, we designed a self-evolving policy optimization scheme. Specifically, during the cold start phase, the policy model π_θ serves as its own teacher model. In each iteration, we first employ π_θ to curate cold start data and perform supervised fine-tuning on itself, and then proceed with reinforcement learning. This process is repeated over multiple sequential iterations. Later cold-start sets are therefore generated by the updated policy rather than by a fixed external teacher.

Table 5: Experimental results from the first five iterations of the self-evolution process.

Methods	AIME 25	Math	Olympiad	GPQA
HDPO (Iter 1)	29.32	90.15	61.98	67.04
HDPO (Iter 2)	30.46	90.77	62.84	68.15
HDPO (Iter 3)	31.63	91.75	63.96	69.23
HDPO (Iter 4)	31.32	92.33	63.85	69.06
HDPO (Iter 5)	31.43	92.14	63.89	69.48

As presented in Table 5, scores rise through the third iteration on all four benchmarks, with smaller and non-monotonic changes thereafter. After five iterations, the model's performance is almost comparable to that of using `Qwen3-235B-A22B-Instruct-2507` as the teacher model to construct cold-start data. This indicates that our HDPO can also stimulate the reasoning ability of LLMs through a self-evolving iterative loop, thereby offering a scalable approach to continuously optimize the capability of the policy model.

Appendix B to Appendix I examine diversity measurement, diversity–accuracy trade-offs, model choice, optimization algorithms, candidate-set size, inference latency, alternative reasoning strategies, and comparisons with prior work.

5 Conclusion

In this paper, we propose Hint-Guided Diversified Policy Optimization (HDPO) to broaden the solution space explored during LLM reasoning. HDPO follows a "propose-select-think" process, generating candidate hints, selecting a promising solution, and developing a reasoning trace. Diversity scheduling and a reliability reward guide optimization toward distinct and reliable solutions. Experiments across multiple benchmarks show improved LLM reasoning. The self-evolution results further show that HDPO can refine its own cold-start data without relying exclusively on cold-start examples supplied by a stronger teacher model.

AI USE STATEMENT

We used generative AI tools solely to improve the manuscript's clarity, fluency, and academic style. All suggested revisions were carefully reviewed to ensure that the original technical content and intended meaning were fully preserved.

ETHICS STATEMENT

This study was conducted in accordance with established ethical research standards. It relies exclusively on publicly available reasoning benchmarks and pretrained language models, all of which are appropriately cited in the paper.

REPRODUCIBILITY STATEMENT

The HDPO method and reward definitions are described in Section 3, while Section 4 specifies the evaluation setting, baselines, and ablations. Appendix A provides data-construction and training details, and Appendix J presents the structured-reasoning prompt. These descriptions support replication of the reported experiments.

REFERENCES

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

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 17682–17690, 2024.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde, Jared Kaplan, Harri Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

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

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. RLAIF vs. RLHF: scaling reinforcement learning from human feedback with AI feedback. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=uydQ2W41KO.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In 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 Information 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/18abbeef8cfe9203fdf9053c9c4fe191-Abstract-Conference.html.

Che Liu, Haozhe Wang, Jiazhen Pan, Zhongwei Wan, Yong Dai, Fangzhen Lin, Wenjia Bai, Daniel Rueckert, and Rossella Arcucci. Beyond distillation: Pushing the limits of medical llm reasoning with minimalist rule-based rl. *arXiv preprint arXiv:2505.17952*, 2025a.

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

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

Renjie Luo, Zichen Liu, Xiangyan Liu, Chao Du, Min Lin, Wenhu Chen, Wei Lu, and Tianyu Pang. Language models can learn from verbal feedback without scalar rewards. *arXiv preprint arXiv:2509.22638*, 2025b.

MAA. American mathematics competitions (AMC 10/12). Mathematics Competition Series, n.d. URL https://maa.org/math-competitions/amc.

math ai. AIME 2025 dataset. https://huggingface.co/datasets/math-ai/aime25, 2025.

Maxwell-Jia. AIME 2024 dataset. https://huggingface.co/datasets/Maxwell-Jia/AIME_2024, 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 feedback. 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 Information 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/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

Sangwon Ryu, Heejin Do, Yunsu Kim, Gary Lee, and Jungseul Ok. Multi-dimensional optimization for text summarization via reinforcement learning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5858–5871, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.319. URL https://aclanthology.org/2024.acl-long.319/.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017. URL http://arxiv.org/abs/1707.06347.

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

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y Charles, et al. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July*

21-27, 2024, Proceedings of Machine Learning Research, pp. 50622–50649. PMLR / OpenReview.net, 2024a. URL https://proceedings.mlr.press/v235/wang24z.html.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024b.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. 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 Information 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/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Haoyuan Wu, Xueyi Chen, Rui Ming, Jilong Gao, Shoubo Hu, Zhuolun He, and Bei Yu. Totrl: Unlock LLM tree-of-thoughts reasoning potential through puzzles solving. *CoRR*, abs/2505.12717, 2025. doi: 10.48550/ARXIV.2505.12717. URL https://doi.org/10.48550/arXiv.2505.12717.

Tianbao Xie, Siheng Zhao, Chen Henry Wu, Yitao Liu, Qian Luo, Victor Zhong, Yanchao Yang, and Tao Yu. Text2reward: Reward shaping with language models for reinforcement learning. 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=tUM39YTRxH.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*, 2025.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.

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

Xingjian Zhang, Siwei Wen, Wenjun Wu, and Lei Huang. Edge-grpo: Entropy-driven grpo with guided error correction for advantage diversity. *arXiv preprint arXiv:2507.21848*, 2025b.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025c.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=WZH7099tgfM.

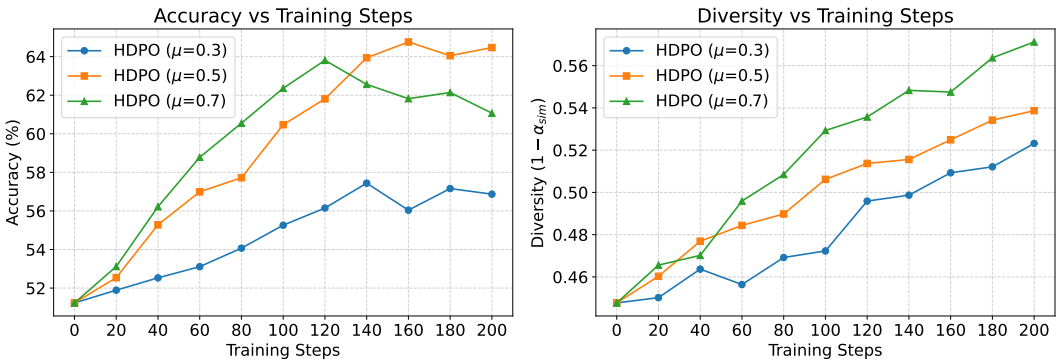


Figure 5: The effect of different weights of diversity rewards on model performance for OlympiadBench.

A Implementation Details

For the data construction in the cold start phase, we sample $L = 10$ answers each time to evaluate reliability. After filtering, 83,279 data entries were retained for supervised fine-tuning. During the reinforcement learning phase, we sample $G = 8$ responses in each rollout, with the maximum length of the prompt set to 1024 and the maximum length of the response set to 3072. To calculate the diversity reward, we employ `Qwen3-Embedding-0.6B` (Zhang et al., 2025c) as π_s to encode each candidate solution and set $\mu = 0.5$ to control the degree of diversity. The maximum training steps t_{max} is set to 200, and the number of warm-up steps t_{wp} before applying the diversity reward is set to 20. All our experiments were conducted on a server equipped with 8 $\times$ NVIDIA A100 (80GB) GPUs.

B Solution Diversity Measurement

To improve computational efficiency, HDPO employs `Qwen3-Embedding-0.6B` to encode candidate solutions and calculates cosine similarity to measure the diversity of solutions. To further investigate alternative approaches, we also analyzed encoding with `Qwen3-Embedding-4B` and `bge-m3` (Chen et al., 2024), as well as measuring similarity by calculating the sum of pairwise BLEU_1 and ROUGE_l scores (i.e., HDPO (B&R)). As shown in Table 6, the more powerful the embedding model, the more beneficial it is for model training, attributed to the fact that the embedding model directly influences the feedback for the diversity of candidate solutions generated by the model. Additionally, the method based on lexical matching does not take into account semantic-level information, demonstrate significantly poorer performance in diversity measurement compared to embedding-based approaches.

Table 6: Comparison of different diversity measurement methods.

Methods	AIME 25	Math	Olympiad	GPQA
HDPO (0.6B)	31.69	93.28	64.76	69.89
HDPO (4B)	32.12	93.85	65.54	70.26
HDPO (bge-m3)	31.24	92.73	64.48	69.31
HDPO (B&R)	29.76	92.01	62.81	67.76
HDPO (LLM-Div)	32.51	92.14	64.22	70.47

Another optional approach is to prompt the LLM to determine the similarity between candidate solutions. Specifically, we prompt `Qwen3-4B-Instruct-2507` to determine the similarity between two candidate solutions. The prompt we use is shown in Figure 7. Next, we parse out the probability that the model outputs "Yes" as a measure of the similarity between the two candidate solutions. As shown by "HDPO (LLM-Div)" in Table 6, the performance of this approach is comparable to that of using the embedding representation to measure the semantic similarity. This indicates that semantic similarity can effectively reflect the correlation between solutions. However, the method of using the LLM to judge the relevance of solutions will lead to more computational overhead, so we adopt the approach of calculating the semantic similarity based on the embedding representation.

C Trade-off between Diversity and Accuracy

To prevent reward hacking caused by excessively low or high diversity, we control the diversity level through the hyperparameter μ. We visualize the changes in accuracy and diversity during model training at different levels of diversity by setting different values of μ, as shown in Figure 5. We can observe that both lower ($\mu = 0.3$) and higher ($\mu = 0.7$) levels of diversity cause instability in model training, while the best results are achieved at $\mu = 0.5$. When the diversity level is low, the model struggles to discern distinctions among candidate solutions, potentially generating homogeneous solutions. Conversely, excessive diversity induces an overemphasis on solution heterogeneity, causing the optimization process to prioritize the diversity reward at the expense of solution correctness and thereby precipitating reward hacking behavior.

D Base Model vs. Instruction-tuned Model

In accordance with the guidelines provided by the official Qwen2.5-Math repository, the initial analysis employed the base model Qwen2.5-Math-7B, which serves as a better starting point for fine-tuning. To investigate the generalizability of HDPO to instruction-tuned architectures, we further utilized Qwen2.5-Math-7B-Instruct as the backbone model. As demonstrated in Table 7, training with HDPO yields significant improvements in the reasoning performance of the instruction-tuned variant.

Table 7: Performance comparison of GRPO and HDPO with Qwen2.5-Math-7B-Instruct.

Variant	AIME 25	Math	Olympiad	GPQA
GRPO	25.72	91.36	59.34	64.51
HDPO	33.14	94.17	67.18	72.45

E Applying to Alternative Policy Optimization Algorithms

Table 8: Experimental results using Dr.GRPO, DAPO, and GSPO as baseline algorithms.

Methods	AMC 23	AIME 24	AIME 25	Math	Minerva	Olympiad	GPQA	Avg.
Dr.GRPO	81.80	42.53	23.16	88.72	44.24	52.07	59.74	56.04
HDPO-Dr.GRPO	84.75	52.83	31.55	93.14	55.55	58.94	67.67	63.49
DAPO	80.17	41.46	26.54	89.81	46.85	54.02	57.75	56.66
HDPO-DAPO	87.22	50.42	32.25	94.33	54.61	59.98	68.73	63.93
GSPO	79.09	42.42	26.31	91.08	43.63	52.57	60.94	55.15
HDPO-GSPO	85.31	51.53	37.32	93.09	53.62	56.52	68.14	63.65

Since HDPO is agnostic to the specific policy optimization algorithm, it can be readily adapted to other algorithms. To demonstrate this flexibility, we selected Dr.GRPO (Liu et al., 2025b), DAPO (Yu et al., 2025), and GSPO (Zheng et al., 2025) as representative baseline algorithms for evaluation. As shown in Table 8, integrating HDPO into each baseline algorithm yielded consistent performance improvements, thereby establishing the generalizability of our approach across diverse policy optimization frameworks.

F Impact of Maximum Number of Candidates

To enhance the diversity of solutions explored by the model, we allow it to generate up to $\mathcal{M} = 5$ candidate solutions. We systematically investigate the influence of varying the maximum number of candidates $\mathcal{M}$ on model performance, with results presented in Table 9. The empirical analysis reveals that both insufficient and excessive values of $\mathcal{M}$ adversely affect performance. When $\mathcal{M}$ is too small, the model's exploration of the solution space remains constrained, increasing the likelihood of missing valid solutions. Conversely, when $\mathcal{M}$ is overly large, the inclusion of redundant or low-quality candidates introduces noise that can hinder the model's ability to identify the correct answer. An optimal trade-off between coverage and conciseness is achieved at $\mathcal{M} = 5$, which yields the best overall performance.

Table 9: The impact of different maximum candidate numbers $\mathcal{M}$ on model performance.

$\mathcal{M}$	AIME 25	Math	Olympiad	GPQA
2	27.31	91.39	58.95	65.43
3	28.46	92.85	60.26	68.26
4	30.18	93.72	62.48	69.31
5	**31.69**	93.28	**64.76**	**69.89**
6	30.84	**93.85**	64.19	68.73
7	29.36	92.61	62.74	68.16

G Inference Latency Analysis

Table 10: Analysis of the trade-off between performance and efficiency.

Variant	AIME 25	Math	Olympiad	GPQA
EDGE-GRPO	19.37 (0.30)	84.10 (0.07)	48.79 (0.08)	46.76 (0.14)
Oat	16.64 (0.39)	85.34 (0.07)	50.82 (0.10)	44.43 (0.13)
SimpleRL	16.34 (0.58)	85.22 (0.09)	49.84 (0.12)	47.31 (0.13)
LUFFY	26.61 (0.72)	87.99 (0.14)	56.09 (0.18)	61.51 (0.24)
Critique-GRPO	28.44 (0.88)	91.22 (0.19)	61.46 (0.26)	67.28 (0.33)
GRPO	24.55 (0.64)	89.72 (0.12)	53.68 (0.18)	60.18 (0.17)
HDPO	31.69 (0.82)	93.28 (0.16)	64.76 (0.22)	69.89 (0.28)

To further analyze the trade-off between computational resource consumption and performance, we took Qwen2.5-Math-7B as an example to analyze the inference latency. As shown in Table 10, the model performance is reported outside the parentheses, while the average inference latency (in seconds) per sample is reported inside the parentheses. It can be observed that HDPO increases the inference latency by less than 0.2s compared to previous SOTA methods, while achieving significant performance improvements. Additionally, the inference delay of HDPO is within 1 second, making it deployable in real-world scenarios. While the computational overhead of HDPO is a legitimate concern, it represents a principled trade-off between computational cost and reasoning quality.

H Comparison with Other Advanced Reasoning Strategies

To further validate the effectiveness of HDPO, we compared it against Self-Consistency (SC) (Wang et al., 2023) and ToTRL (Wu et al., 2025), as detailed in Table 11. For the SC baseline, we generated 40 reasoning paths per instance and employed majority voting to determine the final answer. The empirical results indicate that SC underperforms relative to both ToTRL and HDPO. This limitation likely stems from inadequate exploration during the inference phase, which renders the method susceptible to convergence on erroneous reasoning traces. Furthermore, HDPO

Table 11: Performance comparison of SC, ToTRL, and HDPO with Qwen2.5-Math-7B.

Method	AIME 25	Math	Olympiad	GPQA
SC	19.73	85.34	48.56	54.12
ToTRL	26.54	90.36	57.82	63.24
HDPO	31.69	93.28	64.76	69.89

demonstrates significantly superior performance compared to ToTRL. While ToTRL prioritizes answer correctness, it lacks feedback regarding the solution process. This absence of process-level supervision constrains the model's ability to explore optimal solution paths.

I POSITIONING OF HDPO

Policy Internalization vs. Inference-Time Search. Unlike Tree-of-Thought (ToT) (Yao et al., 2023) and Self-Consistency (Wang et al., 2023) paradigms that rely on repeated model calls, explicit tree expansion, or post-hoc voting during inference, HDPO internalizes the "explore–evaluate–select" cognitive cycle directly into the policy network. Through structured reinforcement learning, the model learns to generate diverse candidate hints and autonomously select the most reliable trajectory in a **single forward pass**. This design preserves the fault-tolerance benefits of multi-path reasoning while eliminating the linear-to-exponential latency overhead inherent to search-based methods.

Explicit Optimization for Diversity and Reliability. Departing from accuracy-centric frameworks such as LUFFY (Yan et al., 2025) and Critique-GRPO (Zhang et al., 2025a), which primarily optimize for outcome-level correctness or rely on external process critiques, HDPO directly addresses solution homogenization and unreliable trajectory selection. Our pipeline explicitly internalizes diversity and reliability into the policy: (1) *Cold Start for Structured Reasoning* establishes the "propose-select-think" trajectory via SFT, rigorously filtering data by both answer correctness and *candidate reliability* (validated through lightweight accuracy sampling). (2) *Hint-Guided Diversified RL* injects dual optimization signals absent in prior frameworks: a sinusoidally scheduled *diversity reward* penalizes semantic redundancy to prevent homogeneous convergence, while a confidence-based *reliability reward* (proxied by token entropy) trains the model to autonomously rank and select the most robust candidate. By embedding these objectives directly into training, HDPO transcends correctness-centric paradigms, yielding a policy that inherently maintains broad solution coverage and fault-tolerant selection in a single forward pass.

Synergy between Diversity and Reliability. The diversity reward (r_{div}) and reliability reward (r_{rel}) operate on *decoupled* reasoning phases: r_{div} encourages variation during candidate *proposal*, while r_{rel} guides the *selection* step. Crucially, r_{div} is gated by outcome correctness, preventing exploration from compromising accuracy. The diversity scheduling strategy further ensures that diversity optimization only activates after the warm-up phase, allowing the model to first master structural compliance and baseline correctness.

Insights on Entropy as a Reliability Proxy. While the selection step initially relies on entropy-based confidence to choose a hint for further reasoning, the ultimate optimization signal is the Verifiable Reward (outcome correctness). If the model selects a hint with low entropy (high confidence) that leads to an incorrect solution, the RLVR mechanism assigns an accuracy reward of zero. Over time, the policy learns to associate specific reasoning patterns with correctness, effectively calibrating the selection policy through trial and error, even if the raw token entropy remains uncalibrated.

Moreover, by incentivizing the generation of multiple candidate outlines before selection, the model reduces the risk of a single overconfident error dominating the process. Even if the entropy metric misranks the candidates, the correct reasoning path is more likely to be present in the candidate set compared to standard single-pass reasoning. The RL objective then reinforces the selection of hints that lead to verifiable success, gradually aligning the reliability signal with actual performance rather than raw entropy.

J PROMPT FOR STRUCTURED REASONING

We provide the prompt used for data construction in the cold start phase in Figure 6. We default to allowing the teacher model to generate up to $\mathcal{M} = 5$ candidate solutions. Each solution overview includes 1 to 5 sentences, containing high-level solution strategies and concepts.

Structured Reasoning Prompt

```
Given a problem, please first consider all possible candidate
solutions, and place an overview of all candidate solutions within
<Candidate Solutions></Candidate Solutions>.  Different solution
overviews should be separated by "\n\n".

1.  The overview of the solution should be as concise as possible,
containing 1 to 5 sentences.  It should only elaborate on the
high-level strategies and concepts, without going into specific
calculations.

2.  Different candidate solutions should exist differences at the
method level, so do not output similar solution overviews.

3.  Do not use line breaks within the same candidate solution
overview, and use "\n\n" to separate different candidate solutions.

4.  You can propose 1 to 5 candidate solutions, each preceded by
the serial number enclosed in square brackets, such as "[1]", "[2]",
"[3]", etc.  Each candidate solution overview is a core summary of
the proposed solution.

After thinking through the candidate solution overviews, choose the
most reliable one and place the step number of the chosen solution
(e.g., "[1]") within <selected></selected>.  Next, proceed with
step-by-step reasoning based on the chosen solution.  Place the
reasoning process in the <thinking></thinking> and put your final
answer in \\boxed{}.

Problem:  {}
```

Figure 6: The prompt used in structured reasoning.

Relevance Judgment Prompt

```
Given a question and its corresponding two candidate solution
outlines, please determine whether these two candidate solutions
are the same.  We consider them to be the same when they meet the
following two conditions.

1.  They use the same branch of mathematics or specific theorem.
2.  They rely on the same underlying logical dependencies, although
the expressions may differ.

If the two candidate solutions are the same, output "Yes", otherwise
output "No", do not output redundant content.

Question:  {question}
Candidate Solution 1:  {candidate solution 1}
Candidate Solution 2:  {candidate solution 2}
```

Figure 7: The prompt used for solution relevance judgment.

K Case Study

To further analyze the advantages of HDPO, we provide three case studies.

Case 1. The first case, illustrated in Figures 8 and 9, highlights a critical limitation of the traditional reasoning approach. As depicted in Figure 8, the traditional reasoning process verifies the constraint

only at three discrete points within the interval $[0, 1]$, despite the requirement that the inequality hold for all $x \in [0, 1]$. Consequently, the model persists along this flawed line of reasoning and ultimately arrives at an incorrect solution. In contrast, HDPO's "propose-select-think" reasoning framework enables it to generate candidate solutions that incorporate Chebyshev polynomials, an essential insight that leverages the extremal properties of these polynomials to correctly determine the maximum value.

Case 2. Figures 10 and 11 present a second illustrative example that demonstrates a different failure mode of the conventional reasoning process. In Figure 10, the model eventually adopts the correct solution in its third step by situating the circle within a coordinate system. However, it has already introduced an error in the second step by erroneously assuming that the distance from point O to chord $\overline{AB}$ equals 4. This error stems from the model's failure to recognize the proper solution strategy at the outset. By contrast, the HDPO method depicted in Figure 11 introduces coordinate geometry as a candidate approach from the beginning and maintains this coherent line of reasoning throughout, ultimately yielding the correct solution.

Case 3 (Failure Case). In addition, we also present a failure case of HDPO, as shown in Figure 12. In this example, the model chose the first candidate solution, but this solution did not take into account the use of complex number theory, which represents the fundamental approach to solving this problem. Consequently, the model made an erroneous assumption during reasoning. Specifically, it incorrectly concluded that angle pairs such as $(8°, 352°)$ and $(16°, 344°)$ possess opposite cosine values, failing to incorporate the complex number perspective into its analysis. The case shows that when the initial set of candidate solutions is generated without the correct answer, the subsequent reasoning is compromised. Therefore, steering the model toward the generation of more pertinent candidate solutions could enhance the reasoning capabilities of LLMs.

Case 1 (Generation Model: Qwen2.5-Math-7B-GRPO)

Question:

Let a, b, c be real numbers such that

$$|ax^2 + bx + c| \le 1$$

for all $0 \le x \le 1$. Find the largest possible value of $|a| + |b| + |c|$.

Response:

<thinking>

To solve the problem, we need to find the maximum value of $|a| + |b| + |c|$ given the constraint $|ax^2 + bx + c| \le 1$ for all $0 \le x \le 1$.

We can start by evaluating the polynomial at specific points within the interval $[0, 1]$. The points $x = 0$, $x = \frac{1}{2}$, and $x = 1$ are good choices because they will give us a system of equations that we can use to determine a, b, and c.

1. At $x = 0$, we have $|c| \le 1$.
2. At $x = 1$, we have $|a + b + c| \le 1$.
3. At $x = \frac{1}{2}$, we have $|\frac{a}{4} + \frac{b}{2} + c| \le 1$.

We need to find the values of a, b, and c that maximize $|a| + |b| + |c|$ while satisfying these constraints. Let's consider the possible values for c. Since $|c| \le 1$, we can set $c = 1$ or $c = -1$ to maximize the sum.

Let's start with $c = 1$. Then we have:

1. $|a + b + 1| \le 1$, which implies $-2 \le a + b \le 0$.
2. $|\frac{a}{4} + \frac{b}{2} + 1| \le 1$, which implies $-2 \le \frac{a}{4} + \frac{b}{2} + 1 \le 0$, or $-3 \le \frac{a}{4} + \frac{b}{2} \le -1$.

We can solve these inequalities to find the values of a and b.

Let's consider the boundary cases to maximize $|a| + |b| + |c|$.

If we set $a = 4$ and $b = -6$, we get: 1. $|4 - 6 + 1| = |-1| = 1 \le 1$, which satisfies the constraint. 2. $|\frac{4}{4} + \frac{-6}{2} + 1| = |1 - 3 + 1| = |-1| = 1 \le 1$, which satisfies the constraint.

So, $|a| + |b| + |c| = |4| + |-6| + |1| = 4 + 6 + 1 = 11$.

Let's check if there are any other possible values that could give a larger sum. If we set $c = -1$, we get:
1. $|a + b - 1| \leq 1$, which implies $0 \leq a + b \leq 2$.
2. $|\frac{a}{4} + \frac{b}{2} - 1| \leq 1$, which implies $0 \leq \frac{a}{4} + \frac{b}{2} \leq 2$, or $0 \leq \frac{a}{4} + \frac{b}{2} \leq 2$.
If we set $a = -4$ and $b = 6$, we get:
1. $|-4 + 6 - 1| = |1| = 1 \leq 1$, which satisfies the constraint.
2. $|\frac{-4}{4} + \frac{6}{2} - 1| = |-1 + 3 - 1| = |1| = 1 \leq 1$, which satisfies the constraint.
So, $|a| + |b| + |c| = |-4| + |6| + |-1| = 4 + 6 + 1 = 11$.
Therefore, the largest possible value of $|a| + |b| + |c|$ is $\boxed{11}$.
</thinking>
$\boxed{11}$ ✘ **Correct Answer: 17**

Figure 8: The output produced by Qwen2.5-Math-7B following GRPO training with standard chain-of-thought reasoning in the first test case.

Case 1 (Generation Model: Qwen2.5-Math-7B-HDPO)

Question:
Let a, b, c be real numbers such that

$$|ax^2 + bx + c| \leq 1$$

for all $0 \leq x \leq 1$. Find the largest possible value of $|a| + |b| + |c|$.

Response:
<Candidate Solutions>
[1] Use the fact that the quadratic polynomial is bounded by 1 in absolute value on [0,1], and evaluate it at key points such as x=0, x=1, and x=0.5 to derive constraints on a, b, c. Then express a, b, c in terms of these values and use linear programming or symmetry arguments to maximize |a| + |b| + |c|.
[2] Consider the polynomial as a function in a finite-dimensional space and use the extreme values of the polynomial on [0,1] to form a system of inequalities. Apply the method of Lagrange multipliers to maximize the linear functional |a| + |b| + |c| subject to the quadratic constraint.
[3] Assume symmetry or specific forms (e.g., a=0, or b=0) to reduce complexity and test boundary cases. However, this may miss the global maximum if the optimal polynomial is not symmetric or has non-zero a and b.
[4] Use Chebyshev polynomials, which are known to be extremal with respect to maximum deviation from zero, and map the standard Chebyshev polynomial on [-1,1] to the interval [0,1]. Scale and shift the Chebyshev polynomial to satisfy the bound $|P(x)| \leq 1$ and compute the coefficients to find the maximum of |a| + |b| + |c|.
[5] Use polynomial interpolation and optimization: since the polynomial is bounded on [0,1], consider the values at three points (e.g., 0, 1/2, 1) and use the resulting system to express coefficients, then apply bounds and optimization techniques to maximize the sum of absolute values.
</Candidate Solutions>
<selected>[4]</selected>
<thinking>
We are given that $|ax^2 + bx + c| \leq 1$ for all $0 \leq x \leq 1$, and we are to find the maximum value of $|a| + |b| + |c|$.
The key insight is to use the properties of Chebyshev polynomials, which are known for being extremal with respect to the maximum norm.

The Chebyshev polynomial of degree 2 on $[-1,1]$ is $T_2(x) = 2x^2 - 1$, which oscillates between -1 and 1 at three points.
We want to map this polynomial to the interval $[0,1]$. Let $t = 2x - 1$, which maps $x \in [0,1]$ to $t \in [-1,1]$. Then $x = \frac{t+1}{2}$, and we can substitute into $T_2(t) = 2t^2 - 1$:

$$P(x) = T_2(2x-1) = 2(2x-1)^2 - 1 = 2(4x^2 - 4x + 1) - 1 = 8x^2 - 8x + 2 - 1 = 8x^2 - 8x + 1.$$

So $a = 8$, $b = -8$, $c = 1$, and $|a| + |b| + |c| = 8 + 8 + 1 = 17$.
Now, we check that $|P(x)| \leq 1$ for $0 \leq x \leq 1$. Since $P(x) = T_2(2x-1)$, and $T_2(t)$ has maximum absolute value 1 on $[-1,1]$, it follows that $|P(x)| \leq 1$ for all $x \in [0,1]$.
Is this the maximum possible? Suppose there exists another polynomial $Q(x) = ax^2 + bx + c$ with $|Q(x)| \leq 1$ on $[0,1]$ and $|a| + |b| + |c| > 17$. By the properties of Chebyshev polynomials, the maximum of $|a| + |b| + |c|$ under the given constraint is achieved at the extremal polynomial, which is a scaled and shifted Chebyshev polynomial. Since $P(x) = 8x^2 - 8x + 1$ achieves $|a| + |b| + |c| = 17$ and satisfies the bound, it must be the maximum.
Thus, the largest possible value is $\boxed{17}$.
</thinking>
$\boxed{17}$ ✓

Figure 9: The output produced by Qwen2.5-Math-7B following HDPO training using "propose-select-think" structured reasoning in the first test case.

Case 2 (Generation Model: Qwen2.5-Math-7B-GRPO)

Question:
Circle O has radius 6. Point P lies outside circle O, and the shortest distance from P to circle O is 4. Chord $\overline{AB}$ is parallel to $\overleftrightarrow{OP}$, and the distance between $\overline{AB}$ and $\overleftrightarrow{OP}$ is 2 . Compute $PA^2 + PB^2$.

Response:
<thinking> To solve the problem, we need to find the value of $PA^2 + PB^2$. Here's the step-by-step reasoning:
1. **Understand the Geometry:** - Circle O has a radius of 6.
- Point P is 4 units away from the circle, so the distance from P to the center O is $6 + 4 = 10$.
- Chord $\overline{AB}$ is parallel to $\overleftrightarrow{OP}$ and the distance between $\overline{AB}$ and $\overleftrightarrow{OP}$ is 2.
2. **Find the Length of Chord $\overline{AB}$:**
- Since $\overline{AB}$ is parallel to $\overleftrightarrow{OP}$ and the distance between them is 2, the perpendicular distance from O to $\overline{AB}$ is $6 - 2 = 4$ or $6 + 2 = 8$. But since P is 4 units away from the circle and $\overline{AB}$ is parallel to $\overleftrightarrow{OP}$, the perpendicular distance from O to $\overline{AB}$ must be 4 (as 8 would place $\overline{AB}$ outside the circle).
- Using the Pythagorean theorem in the right triangle formed by the radius, the perpendicular distance, and half the chord, we get:

$$\left(\frac{AB}{2}\right)^2 = 6^2 - 4^2 = 36 - 16 = 20 \implies \frac{AB}{2} = \sqrt{20} = 2\sqrt{5} \implies AB = 4\sqrt{5}$$

3. **Find the Coordinates:**
- Place the circle centered at $O(0,0)$ and point P at $(10,0)$.

- The chord $\overline{AB}$ is parallel to the x-axis and 2 units below the x-axis (since the distance from O to $\overline{AB}$ is 4).
- So, the coordinates of A and B can be $(-2\sqrt{5}, -2)$ and $(2\sqrt{5}, -2)$ respectively.
4. **Calculate PA^2 and PB^2:**
- Using the distance formula, we get:

$$PA^2 = (10-(-2\sqrt{5}))^2+(0-(-2))^2 = (10+2\sqrt{5})^2+2^2 = 100+40\sqrt{5}+20+4 = 124+40\sqrt{5}$$

$$PB^2 = (10-2\sqrt{5})^2+(0-(-2))^2 = (10-2\sqrt{5})^2+2^2 = 100-40\sqrt{5}+20+4 = 124-40\sqrt{5}$$

- Adding PA^2 and PB^2:

$$PA^2 + PB^2 = (124+40\sqrt{5}) + (124-40\sqrt{5}) = 248$$

So, the final answer is:

$$\boxed{248}$$

</thinking>

$\boxed{248}$ ✘ **Correct Answer: 272**

Figure 10: The output produced by Qwen2.5-Math-7B following GRPO training with standard chain-of-thought reasoning in the second test case.

Case 2 (Generation Model: Qwen2.5-Math-7B-HDPO)

Question:
Circle O has radius 6. Point P lies outside circle O, and the shortest distance from P to circle O is 4. Chord $\overline{AB}$ is parallel to $\overleftrightarrow{OP}$, and the distance between $\overline{AB}$ and $\overleftrightarrow{OP}$ is 2 . Compute $PA^2 + PB^2$.

Response:
<Candidate Solutions>
[1] Use geometric symmetry and properties of circles and chords. Since AB is parallel to OP and the distance between them is 2, the perpendicular distance from O to AB is either $4 + 2 = 6$ or $4 - 2 = 2$. This approach relies on parameterizing the circle and solving systems of equations.
[2] Use vector geometry: represent points A and B on the circle with AB parallel to vector $\vec{OP}$, and the perpendicular distance between the lines is 2.
[3] Use coordinate geometry by placing the center of circle O at the origin and point P along the x-axis. Since chord AB is parallel to $\overleftrightarrow{OP}$, it is horizontal, and the distance between AB and $\overleftrightarrow{OP}$ is 2, so AB is either at $y = 2$ or $y = -2$.
</Candidate Solutions>
<selected>[3]</selected>
<thinking>
We place the center of circle O at the origin $(0,0)$. Since the radius is 6, the circle has equation $x^2 + y^2 = 36$.
The shortest distance from point P to the circle is 4. Since the radius is 6, this means that the distance from O to P is $6 + 4 = 10$. We can place P at $(10,0)$, so that it lies on the x-axis.
The line $\overleftrightarrow{OP}$ is the x-axis (since $O = (0,0)$, $P = (10,0)$). The chord AB is parallel to $\overleftrightarrow{OP}$, so it is horizontal. The distance between AB and $\overleftrightarrow{OP}$ is 2. Since AB could be above or below the x-axis, we

consider $y = 2$ and $y = -2$. By symmetry, the value of $PA^2 + PB^2$ will be the same in both cases, so we can choose $y = 2$.
Now, find the points A and B where $y = 2$ intersects the circle $x^2 + y^2 = 36$:

$$x^2 + 2^2 = 36 \Rightarrow x^2 + 4 = 36 \Rightarrow x^2 = 32 \Rightarrow x = \pm\sqrt{32} = \pm 4\sqrt{2}$$

So, $A = (4\sqrt{2}, 2)$, $B = (-4\sqrt{2}, 2)$
Now compute PA^2 and PB^2, where $P = (10, 0)$
First, PA^2:

$$PA^2 = (10 - 4\sqrt{2})^2 + (0 - 2)^2 = (10 - 4\sqrt{2})^2 + 4$$

Expand:

$$(10 - 4\sqrt{2})^2 = 100 - 80\sqrt{2} + 16 \cdot 2 = 100 - 80\sqrt{2} + 32 = 132 - 80\sqrt{2}$$

So:

$$PA^2 = 132 - 80\sqrt{2} + 4 = 136 - 80\sqrt{2}$$

Now PB^2:

$$PB^2 = (10 + 4\sqrt{2})^2 + (0 - 2)^2 = (10 + 4\sqrt{2})^2 + 4$$

Expand:

$$(10 + 4\sqrt{2})^2 = 100 + 80\sqrt{2} + 32 = 132 + 80\sqrt{2}$$

So:

$$PB^2 = 132 + 80\sqrt{2} + 4 = 136 + 80\sqrt{2}$$

Now add $PA^2 + PB^2$:

$$(136 - 80\sqrt{2}) + (136 + 80\sqrt{2}) = 136 + 136 = 272$$

The $\sqrt{2}$ terms cancel out.
Thus, the answer is $\boxed{272}$
</thinking>
$\boxed{272}$ ✓

Figure 11: The output produced by Qwen2.5-Math-7B following HDPO training using "propose-select-think" structured reasoning in the second test case.

Case 3 (Generation Model: Qwen2.5-Math-7B-HDPO)

Question:
Compute $\sin^2 4^\circ + \sin^2 8^\circ + \sin^2 12^\circ + \cdots + \sin^2 176^\circ$.

Response:
<Candidate Solutions>
[1] Use the identity $\sin^2\theta = \frac{1-\cos 2\theta}{2}$ to rewrite each term in the sum. Then apply the sum of cosine series formula for angles in arithmetic sequence, simplifying the resulting expression using symmetry and periodicity of the cosine function.
[2] Interpret the sum as a discrete approximation of an integral of $\sin^2 x$ over a interval, and use numerical methods or symmetry to estimate the value. However, this approach lacks precision and relies on approximation, making it unsuitable for exact computation.
[3] Use complex exponentials to express $\sin^2\theta$ and convert the sum into a sum of powers of roots of unity, then apply properties of Gauss sums or Fourier analysis on finite groups.
[4] Use a recursive or computational algorithm to evaluate each term numerically and sum them, but this is impractical for exact mathematical reasoning and lacks elegance and precision.
</Candidate Solutions>

<selected>[1]</selected>
<thinking>
We are given the sum:

$$\sin^2 4^\circ + \sin^2 8^\circ + \sin^2 12^\circ + \cdots + \sin^2 176^\circ$$

This is a sum of $\sin^2\theta$ for $\theta = 4k^\circ$ where $k = 1, 2, \ldots, 44$, since $4 \times 44 = 176$.
We use the identity:

$$\sin^2\theta = \frac{1 - \cos 2\theta}{2}$$

Applying this to each term:

$$\sin^2(4k^\circ) = \frac{1 - \cos(8k^\circ)}{2}$$

So the entire sum becomes:

$$\sum_{k=1}^{44} \sin^2(4k^\circ) = \sum_{k=1}^{44} \frac{1 - \cos(8k^\circ)}{2} = \frac{1}{2}\sum_{k=1}^{44} 1 - \frac{1}{2}\sum_{k=1}^{44} \cos(8k^\circ)$$

$$= \frac{1}{2} \cdot 44 - \frac{1}{2}\sum_{k=1}^{44} \cos(8k^\circ) = 22 - \frac{1}{2}\sum_{k=1}^{44} \cos(8k^\circ)$$

Now we need to compute:

$$\sum_{k=1}^{44} \cos(8k^\circ)$$

The sum of $\cos(8k^\circ)$ for $k = 1$ to 44 can be evaluated using symmetry. Note that the angles $8k^\circ$ for $k = 1$ to 44 are evenly spaced from 8° to 352°, which is half the circle (since $8 \times 44 = 352$). These angles are symmetric about 180°, and the cosine function is even but symmetric about 180°, so the sum of cosines over a symmetric arithmetic sequence about 180° with evenly spaced steps is zero. To see this: the sequence $8^\circ, 16^\circ, \ldots, 352^\circ$ is symmetric about 180°, and $\cos(360^\circ - x) = \cos x$, so pairs like $(8^\circ, 352^\circ)$, $(16^\circ, 344^\circ)$, etc., have equal and opposite cosines. Since there are 44 terms (even number), they can be perfectly paired, and each pair sums to zero.
Therefore:

$$\sum_{k=1}^{44} \cos(8k^\circ) = 0$$

So the original sum is:

$$22 - \frac{1}{2} \cdot 0 = 22$$

</thinking>
$\boxed{22}$ ✘ **Correct Answer:** $\frac{45}{2}$

Figure 12: A failure case of Qwen2.5-Math-7B following HDPO training, where the candidate solutions did not include the most reliable solution.