Exploring the Limit of Outcome Reward for Learning Mathematical Reasoning

Chengqi Lyu[1*] **Songyang Gao**[1*] **Yuzhe Gu**[1,2*] **Wenwei Zhang**[1*†] **Jianfei Gao**[1]
Kuikun Liu[1] **Ziyi Wang**[1] **Shuaibin Li**[1] **Qian Zhao**[1] **Haian Huang**[1]
Weihan Cao[1] **Jiangning Liu**[1] **Hongwei Liu**[1] **Junnan Liu**[1] **Songyang Zhang**[1]
Dahua Lin[1,3,4] **Kai Chen**[1†]
[1]Shanghai AI Laboratory [2]Shanghai Jiao Tong University
[3]MMLab, The Chinese University of Hong Kong [4]HKGAI under InnoHK
`{lvchengqi,gaosongyang,guyuzhe,zhangwenwei,chenkai}@pjlab.org.cn`

Abstract

Reasoning abilities, especially those for solving complex math problems, are crucial components of general intelligence. Recent advances by proprietary companies, such as o-series models of OpenAI, have made remarkable progress on reasoning tasks. However, the complete technical details remain unrevealed, and the techniques that are believed certainly to be adopted are only reinforcement learning (RL) and the long chain of thoughts. This paper proposes a new RL framework, termed OREAL, to pursue the performance limit that can be achieved through **O**utcome **RE**w**A**rd-based reinforcement **L**earning for mathematical reasoning tasks, where only binary outcome rewards are easily accessible. We theoretically prove that behavior cloning on positive trajectories from best-of-N (BoN) sampling is sufficient to learn the KL-regularized optimal policy in binary feedback environments. This formulation further implies that the rewards of negative samples should be reshaped to ensure the gradient consistency between positive and negative samples. To alleviate the long-existing difficulties brought by sparse rewards in RL, which are even exacerbated by the partial correctness of the long chain of thought for reasoning tasks, we further apply a token-level reward model to sample important tokens in reasoning trajectories for learning. With OREAL, for the first time, a 7B model can obtain 94.0 pass@1 accuracy on MATH-500 through RL, being on par with 32B models. OREAL-32B also surpasses previous 32B models trained by distillation with 95.0 pass@1 accuracy on MATH-500. Our investigation also indicates the importance of initial policy models and training queries for RL. Code, models, and data are available at `https://github.com/InternLM/OREAL`.

1 Introduction

Solving complex problems with reasoning capability forms one of the cornerstones of human cognition - a cognitive ability that artificial general intelligence must ultimately master (Xu et al., 2025; Zhong et al., 2024). Among various problem domains, the mathematical problem emerges as a particularly compelling experimental paradigm for AI research (Liu et al., 2023; Matzakos et al., 2023; Ying et al., 2024; Shao et al., 2024), owing to its relatively well-defined structure and availability of precise binary correctness feedback based on the verifiable final answers.

Recent advances in large language models (LLMs) have achieved remarkable progress in mathematical reasoning by the chain-of-thought technics (Wei et al., 2022; Wang et al., 2022;

* Equal contribution † Corresponding author

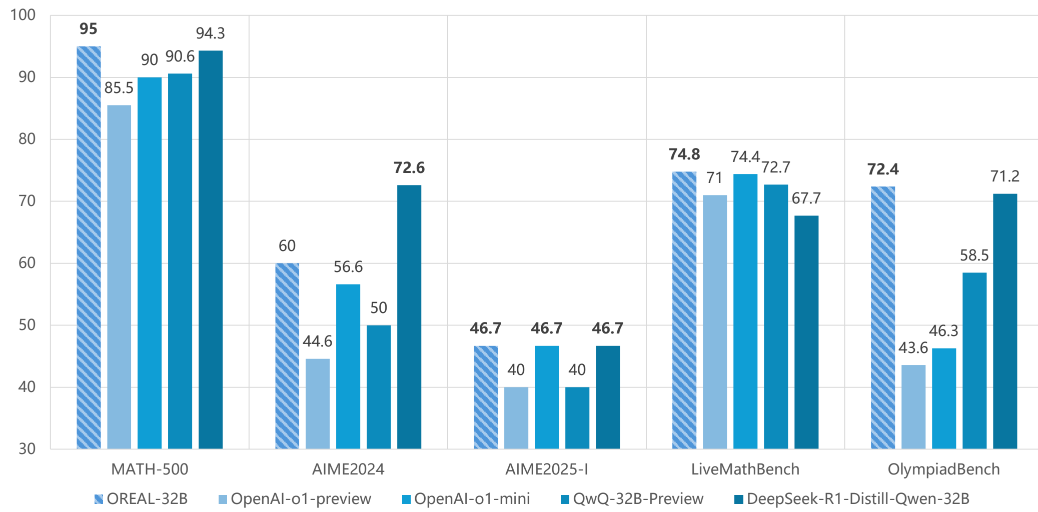

Figure 1: Overall performance between OREAL-32B and some competitive baselines.

Kojima et al., 2022), in which the LLMs are elicited to produce a series of intermediate reasoning steps before providing the final answers to the problem. However, as most of the capable models (*e.g.*, the o-series models by OpenAI (OpenAI, 2024b)) are developed by proprietary companies, there is no clear pathway to develop state-of-the-art reasoning models. Some recent work shows that distillation (Huang et al., 2024; Min et al., 2024) is sufficient to obtain high performance given the accessibility to existing best or near best AI models, reinforcement learning (RL) is believed to be a more fundamental approach and has exhibited potential (DeepSeek-AI et al., 2025) to advance beyond the intelligence boundary of current AI models, using the most capable open-source foundation models (DeepSeek-V3-base (Liu et al., 2024a), *inter alia*).

However, fundamental challenges of sparse reward in RL persist and are even exacerbated in mathematical reasoning tasks that mainly rely on the chain of thought technics with LLMs (Wei et al., 2022): the evaluation of intermediate reasoning steps is labor intensive (Lightman et al., 2023) and its accurate automation approach is still under-explored, thus, the only reliable reward is based on the outcome (correctness of final answer), which is inherently binary and sparse when faced with more than 2000 tokens in the long reasoning trajectories (DeepSeek-AI et al., 2025; Team et al., 2025). Existing approaches have attempted to estimate the advantages or values of reasoning steps by search (Wang et al., 2024; Kazemnejad et al., 2024) or value function-based credit assignment (Schulman et al., 2017; Cui et al., 2025), yet, their performance remains unsatisfactory in comparison with the distilled models (DeepSeek-AI et al., 2025).

This paper aims to conquer the above challenges and proposes a simple framework, termed OREAL, to push the limit of **O**utcome **RE**w**A**rd-based reinforcement **L**earning for mathematical reasoning tasks. OREAL is grounded in the unique characteristics of mathematical reasoning tasks that binary outcome feedback creates an environment where all positive trajectories are equally valid. We first establish that behavior cloning on BoN-sampled positive trajectories is sufficient to achieve KL-regularized optimality, which emerges from the analysis that the positive trajectory from BoN sampling converges to a distribution independent of the sample number. For learning on negative samples, OREAL reveals the necessity of reward shaping to maintain consistent gradient estimation between sampling and target distributions. Such a mechanism compensates for BoN's under-sampling of negative gradients, and enables difficulty-adaptive optimization over both successful and failed trajectories.

Another intrinsic property of mathematical reasoning tasks is the partial correctness in long reasoning chains, which further imposes the learning difficulty of sparse rewards when only a binary outcome reward is available at each iteration of RL training. Thus, OREAL adopts a lightweight credit assignment scheme through a token-level reward model trained using outcome rewards. This mechanism automatically estimates step-wise importance weights by decomposing trajectory advantages, enabling focused learning of critical reasoning steps or errors. The integration of these components yields a theoretically sound framework that effectively bridges the gap between sparse binary feedback and dense policy optimization requirements for mathematical reasoning tasks.

Extensive experimental results show that OREAL effectively improves the mathematical reasoning capability of LLMs. At the 7B parameter scale, to the best of our knowledge, OREAL-7B is the first to obtain the pass@1 accuracy on MATH-500 (Hendrycks et al., 2021) to 91.0 using RL instead of distillation, which even exceeds QwQ-32B-Preview (Team, 2024) and o1-mini (OpenAI, 2024b). OREAL also improves DeepSeek-R1-Distilled-Qwen-7B from 92.8 to 94.0 pass@1 accuracy, being on par with the previous best 32B models. For the 32B model, OREAL-32B outperforms all previous models (Figure 1), both distilled and RL-based, obtaining new state-of-the-art results with 95.0 pass@1 accuracy on MATH-500.

2 Methods

In this section, we first analyze the formulation of RL and the intrinsic properties of underlying binary feedback environments (§2.1), and establish a theoretical foundation for our optimization framework about how to learn from long reasoning chains.

2.1 Preliminary

When adopting a large language model (LLM) for mathematic reasoning, common practices (Shao et al., 2024; Yuan et al., 2023) conduct binary feedback (0/1 reward) based solely on the correctness of LLM's final answer, and perform policy optimization correspondingly.

Policy Optimization Considering a Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ is a finite state space, $\mathcal{A}$ is the action space, $P(s'|s,a)$ specifies the state transition dynamics, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ denotes the discount factor. We focus on KL-regularized policy optimization, which can be formulated as:

$$J(\theta) \triangleq \mathbb{E}_{s\sim\rho_0, a\sim\pi_\theta(\cdot|s)}\left[Q^{\pi_\theta}(s,a)\right] - \alpha \cdot \mathbb{E}_{s\sim\rho_0}\left[D_{\mathrm{KL}}\left(\pi_\theta(\cdot|s) \| \pi_0(\cdot|s)\right)\right] \tag{1}$$

with the state-action value function $Q^{\pi}(s,a) \triangleq \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k r(s_{t+k}, a_{t+k}) \mid s_t = s, a_t = a\right]$ under vanilla policy π.

Best-of-N (BoN) Sampling. As a common and efficient strategy to sample multiple reasoning trajectories from LLMs, Best-of-N sampling selects the trajectory with maximal reward among n independent rollouts from π_0 to enhance policy performance. Formally, given candidate actions $\{a^{(i)}\}_{i=1}^{n} \sim \pi_0(\cdot|s)$, the chosen action is $a^* = \arg\max_{a^{(i)}} Q(s, a^{(i)})$. This strategy effectively leverages the exploration-exploitation trade-off through parallel sampling (Gao et al., 2023; Go et al., 2023).

To quantify the distributional divergence induced by BoN sampling, prior work (Hilton & Gao, 2022; Scheurer et al., 2023; Coste et al., 2023) has analyzed the KL divergence between the BoN distribution π_{BoN} and the original policy π. For continuous trajectory spaces $\mathcal{S}$, the corresponding KL divergence is given by $\mathrm{KL}(\pi_{\mathrm{BoN}} \parallel \pi) = \log n - \frac{n-1}{n}$.

2.2 Learning from Positive Samples

Building upon the reward equivalence principle stated in Eq. A1, we first formalize a key probabilistic characteristic of BoN sampling:

Lemma 2.1. *Let $\pi(\theta, s)$ be a distribution over parameters θ and trajectory s, where each s is associated with a binary reward $R(s) \in \{0, 1\}$. Define $p \triangleq \mathbb{E}_{s\sim\pi(\theta,\cdot)}[R(s) = 1] > 0$. Consider the*

BoN sampling: $n = n_0 \to \infty$ *and sample* $\{s_1, s_2, \ldots, s_n\}$ *i.i.d. from* π_θ*. BoN selects* s^* *uniformly from the subset with* $R(s_i) = 1$*. We have that, The probability of selecting* s^* *is converge to* $\frac{\pi(\theta,s)}{p}$*, which is independent of* n*.*

The proof follows directly from the union law of BoN sampling ($\mathrm{BoN}_{n+m} = \mathrm{BoN}_2(\mathrm{BoN}_m, \mathrm{BoN}_n)$) and the trivial distinguishability of $0-1$ rewards. This result reveals that for problems with attainable positive responses, we are using a BoN generator with an arbitrary sampling budget to construct the positive training samples.

BoNBoN (Gui et al., 2024) empirically shows that BoN sampling achieves the optimal win rate under fixed KL constraint by exhaustive search over the positive support. Therefore, the behavior cloning on BoN-selected positive samples directly learns the analytic solution to Eq. 1. Intuitively, since every correct answer is preferred identically in the outcome-supervised sense, we only need to sample until we get a positive example, whose generating probability distribution will be the same as randomly picking from arbitrarily large numbers of samples.

Based on the theoretical understanding established, we formulate the first component of the learning objective in OREAL by incorporating KL-constrained max-likelihood-objective over positive examples obtained through sampling:

$$\mathcal{L}_1(\theta) = \underbrace{\mathbb{E}_{s\sim\mathcal{D}^+}\left[-\log \pi_\theta(s)\right]}_{\text{Positive example alignment}} + \beta \underbrace{\mathrm{KL}(\pi_\theta \parallel \pi_{\text{old}})}_{\text{Policy constraint}},$$

where $\mathcal{D}^+$ denotes the set of positive trajectories selected via BoN sampling from RL rollouts.

2.3 Learning from Negative Samples

As established in Section 2.2, direct behavioral cloning on positive responses can effectively recover the policy distribution. BOND (Sessa et al., 2024) proposes estimating Jeffreys divergence (Jeffreys, 1946) for the BoN strategy to train with both positive and negative samples, and demonstrates that signals from unsuccessful trajectories provide critical information about decision boundaries and failure modes.

In this section, we will discuss the relationship between the BoN (Best-of-N) distribution and the optimization objective defined in Eq. 1, then elucidate the necessity of reward reshaping when training with negative samples.Specifically, the transformed BoN distribution can be expressed as

$$\pi_{\text{bon}}(s) = \pi(s)\left[R(s)\cdot\frac{1-(1-p)^n}{p} + (1-R(s))\cdot(1-p)^{n-1}\right]. \tag{2}$$

Consider a scenario where two correct and two incorrect solutions are sampled, yielding an empirical accuracy of 50%. However, the probability of selecting negative samples under Best-of-4 becomes $(0.5)^4 = 6.25\%$, significantly lower than the original distribution. This discrepancy necessitates reward shaping to maintain consistency between our optimization target and the expected return under the BoN distribution.

Building on BoN-RLB's (Chow et al., 2024) application of the log-likelihood trick for BoN-aware policy gradients, we analyze the reward shaping technic for negative samples to maintain gradient consistency with Section 2.2. With expectation return p follow the definition in Lemma 2.1. We derive the gradient components as:

$$\mathbb{E}_{s\sim\pi_{\text{bon}}}\left[\nabla_\theta\left(I_{D_+}(s)\log\pi(s)\rho(D_+)\right)\right] = \frac{1}{1-p}\mathbb{E}_{s\sim\pi_{\text{bon}}}\left[\nabla_\theta\left(I_{D_-}(s)\log\pi(s)\rho(D_-)\right)\right],$$

where $I_{D_+}(s)$ and $I_{D_-}(s)$ denote indicator functions for positive and negative sample sets respectively, and $\rho(D) \triangleq \frac{\mathbb{E}_{s\sim\pi_{\text{bon}}}[I_D(s)]}{p_D}$ are their density. Notably, these indicators are independent of policy parameters θ.

This derivation reveals that gradient consistency requires reshaping negative sample rewards to $R^{\star}(s) \triangleq (1-p)R(s)$. Based on this reward shaping, we can construct policy optimization both on positive and negative samples for optimal policy. In this paper, we apply a similar setting to RLOO (Fukunaga & Hummels, 1989), namely $R_{RLOO}(s) = R(s) - \frac{1}{N-1}\sum_{s^{\star}\neq s} R(s^{\star})$ for unbiased mean reward and train with policy gradient. The second part of our OREAL objective is then formulated as below:

$$\mathcal{L}_2(\theta) = \mathbb{E}_{s\sim S_-}\left[F(1-p)\cdot \log \frac{\pi_\theta(s)}{\pi_{old}(s)}\right] + \beta \mathrm{KL}(\pi_\theta \parallel \pi_{\mathrm{old}}),$$

where $p = \mathbb{P}_{\theta\sim\pi}[R(\theta) = 1]$, S_- is the failed subset generated by policy model, and F represents the preprocessing for advantage scores to serve as a generalized form, for example, $F(1-p) \triangleq \frac{r_i - mean(\{r_i ... r_n\})}{std(\{r_i ... r_n\})}$ in the recent GRPO (Shao et al., 2024) algorithm, where $mean(\{r_i ... r_n\}) \to p$ when $n \to \infty$.

2.4 Dealing with Long Reasoning Chains

In the previous discussion, we introduced the adaptation of binary reward training in response space. However, since the outcome supervision only provides feedback at the sequence level, this modeling essentially reduces to a contextual bandit without internal reward modeling within the MDP. We therefore choose to use some low-cost alternatives for sequence-level reweighting.

Since intermediate rewards are not provided in mathematical reasoning tasks, we define an advantage function based solely on outcome feedback:

$$A(s_{\le t}) = V^{\pi}(s_{\le t+1}) - V^{\pi}(s_{\le t}). \tag{3}$$

This formulation treats $A(s_{\le t})$ as a token-wise credit assignment mechanism, estimating each token's contribution toward the final outcome.

For a pair of responses y_1 and y_2 to the same query, their initial values coincide $V_0^1 = V_0^2$. The win rate $p(y_1 > y_2)$ between them then satisfies:

$$\begin{aligned}
\sigma(r(y_1) - r(y_2)) &= \sigma\left(\left(V_0^1 + \sum_{t=0}^{T} \gamma^t A_{y_1}^t\right) - \left(V_0^2 + \sum_{t=0}^{T} \gamma^t A_{y_2}^t\right)\right) \\
&= \sigma\left(\sum_{t=0}^{T} \gamma^t \left(A_{y_1}^t - A_{y_2}^t\right)\right).
\end{aligned} \tag{4}$$

Equation 4 indicates that for any function family $\mathcal{A} = \{A(s_{\le t})\}$, a cumulative reward function through sequence aggregation can be constructed to model rewards: $r^*(s) \triangleq \sum_{t=0}^{T}\gamma^t A(s_{\le t})$, which is trainable via preference pairs $\{(y_w, y_l)\}$ by fitting the outcome feedback.

The learned $A(s_{\le t})$ then serves as a weighting function for credit assignment, which is used to reweight the original training loss, emphasizing critical reasoning steps or errors. Following the practice from (Cobbe et al., 2021), we directly train a token-level reward function $w(s_{\le t})$ satisfying

$$\frac{1}{T}\sum_{t=0}^{T} w(s_{\le t}) = r(s),$$

without constraining KL-divergence to reference model in reward model training. These sequential rewards can serve as a proxy for the contribution of thinking steps to the result accuracy. We further discuss the training details of this model and analyze the visualization of its token-wise scoring effects later in Appendix D.

In practice, we decompose the output weight $w(s)$ for positive and negative samples and clip on the positive axis to prevent reversing the direction of the optimized gradient, denoted as $\omega^+ = \max(2\sigma(w) - 1, 0)$ and $\omega^- \max(1 - 2\sigma(w)$.

With input query d, the overall loss is as follows:

$$\mathcal{L}_{\text{total}}(d) \triangleq \mathbb{E}_{s\sim S}\left[\sum_{t=0}^{T}\left(-\omega^{+}_{s\leq t}\log \pi_\theta(s_{\leq t}|d) I_{D_+}(s) + \eta\ \omega^{-}_{s\leq t}\log\frac{\pi_\theta(s_{\leq t}|d)}{\pi_{old}(s_{\leq t}|d)} I_{D_-}(s)\right)\right] \\ + \beta \mathrm{KL}(\pi_\theta(\cdot|d) \parallel \pi_{\text{old}}(\cdot|d)), \tag{5}$$

where η represents the balancing weights for positive and negative losses.

3 Implementation

3.1 Policy Initialization

We utilize Qwen2.5-7B and Qwen2.5-32B (Yang et al., 2024) as the base model. Initially, we fine-tune the base models using long chain-of-thought data obtained through rejection sampling (Yuan et al., 2023). This rejection sampling fine-tuned (RFT) (Yuan et al., 2023) models then serve as the initialization for the policy model in our RL framework. We also explore to use of DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI et al., 2025) as the initial policy model and perform OREAL on it and discuss the influence of different initial policy models in Appendix C. The training data for the RFT models consists of in-house datasets supported by OpenDataLab (He et al., 2024b) and open-source datasets including Numina (Li et al., 2024) and the training set of MATH (Hendrycks et al., 2021). In addition, we implement a skill-based enhancement approach to further enhance the capability of the model, whose details can be found in Appendix C.1. We report the performance of the initial policy models and discuss the impact of policy initialization in Appendix C.

3.2 Reinforcement Learning

Data Preparation. During the on-policy RL process, we utilize questions from Numina, MATH training sets, and historical AMC/AIME (without AIME2024) competitions. For each question, we independently sample 16 trajectories from the RFT models. To increase the difficulty of training queries, only questions with average correctness rates between 0 and 0.8 are retained for further training.

Outcome Reward Signal. We employ the Qwen2.5-72B-Instruct (Yang et al., 2024) as a generative verifier, combined with a rule-based verifier, to evaluate the correctness of the outputs and provide binary rewards. This combination enhances the robustness of correctness assessment, mitigating issues like the false negative of the rule-based verifier.

Token-level Reward Model. We directly use the binary outcome rewards and optimize using the cross-entropy loss. Details can be found in Appendix D.

Training Algorithm. The loss function for the policy model follows the formulation described in Section 2. The complete RL training procedure is described in Algorithm 1.

Model Training Settings. The policy model is initialized from the RFT model. Similarly, the token-level reward model is also initialized with the same weights, but its output layer is replaced with a linear layer that produces a one-dimensional scalar. The weights of this layer are initialized to zero to ensure unbiased fine-grained token re-weighting at the start of training. The training hyperparameters can be found in can be found in Appendix B.

4 Experiment

4.1 Evaluation Setup

Baseline. We conduct evaluations against several baselines, including GPT-4o-0513 (OpenAI, 2024a), Claude-Sonnet-3.5-1022 (Anthropic, 2024), OpenAI-o1-mini, OpenAI-o1-preview (OpenAI, 2024b), Qwen2.5-Instrust-7B, Qwen2.5-Math-Instrust-7B, Qwen2.5-

Algorithm 1 The OREAL Reinforcement Learning Algorithm

Inputs: Question set $\mathcal{D}$, policy model π_θ, token-level reward model w_θ, number of iterations N, batch size B, number of rollouts per question K.
Initialize policy π_0 and token-level reward model w_0 with π_{sft}.
for $i = 0, \ldots, N$ **do**
  Sample a batch of questions $\mathcal{D}_i \subseteq \mathcal{D}$ of size B.
  For $x \in \mathcal{D}_i$, generate K policy samples: $Y = \{y_1, \ldots, y_K\}$ where $y_k \sim \pi_i(x)$
  Obtain binary rewards $\{r_1, \ldots, r_K\}$ from verifier.
  Compute correctness rate: $p = \frac{1}{K}\sum_{k=1}^{K} r_k$ for reward shaping.
  Retain questions with $0 < p < 1$ to avoid trivial cases.
  Select one correct y^+ and one incorrect sample y^- for each question to avoid imbalance between positive and negative samples.
  Performing fine-grained re-weighting of each token with w_i.
  Use Eq (A2) to update w_i.
  Update π_i with Eq (5)
end for
Return: The optimized policy model π^*.

Instrust-32B (Yang et al., 2024), QwQ-32B-Preview (Team, 2024), DeepSeek-R1-Distill-Qwen-7B, DeepSeek-R1-Distill-Qwen-32B (DeepSeek-AI et al., 2025), SimpleRL (Zeng et al., 2025), PRIME (Cui et al., 2025), rStarMath (Guan et al., 2025). For part of the baseline, we directly use the results from their report, which we mark with *.

Benchmark. We use some well-established mathematical datasets for evaluation, including MATH-500 (Hendrycks et al., 2021), AIME2024 (MAA), AIME2025 (Part1) (MAA), LiveMathBench (Liu et al., 2024c), and OlympiadBench (He et al., 2024a).

Metrics. We use pass@1 as the metric for evaluation under the zero-shot chain-of-thought setting and use greedy decoding for each sample to assess correctness.

4.2 Overall Results

Tabel 1 shows the results of the comprehensive evaluation, highlighting the performance of our proposed models across different parameter scales. Notably, at the 7B scale, OREAL-7B achieves a remarkable pass@1 accuracy of 91.0 on the MATH-500 and 59.9 on OlympiadBench. To the best of our knowledge, this is the first time a model of this size has reached such a high level of accuracy using RL instead of distillation. This performance not only establishes a new milestone for RL-based methods but also surpasses significantly larger models, including QwQ-32B-Preview and OpenAI-o1-mini, demonstrating the effectiveness of our approach. Furthermore, after applying OREAL on the previous best 7B model, DeepSeek-R1-Distill-Qwen-7B, the resulting model, OREAL-DSR1-Distill-Qwen-7B, obtains 94.0 and 66.1 pass@1 accuracy on MATH-500 and OlympiadBench, respectively, setting new records among all 7B models. This result verifies the effectiveness of OREAL even when faced with strong initial policies.

For 32B models, OREAL-32B achieves a groundbreaking pass@1 accuracy of 95.0 on MATH-500, 46.7 on AIME2025-I, 74.8 on LiveMathBench, and 72.4 on OlympiadBench, setting a new state-of-the-art among all previously reported models. These results underscore the advantages of our methodology, including its scalability for training superior mathematical reasoning models across different model sizes.

Compared to the most competitive baseline, DeepSeek-R1-Distill-Qwen series, OREAL-32B demonstrates a clear advantage, whereas OREAL-7B lags slightly behind than the distilled 7B model, despite being trained on the same dataset as OREAL-32B. We attribute this discrepancy to the different affinities of the base models for the post-training data. Qwen-7B and Qwen-32B may exhibit varying degrees of knowledge gaps due to model sizes and pre-training settings. Our training data appears to better complement the existing knowledge of Qwen-32B, while it may be less effective in bridging gaps for Qwen-7B.

Model	MATH-500	AIME2024	AIME2025-I	LiveMath	Olympiad
		API Models			
GPT-4o-1120	72.8	16.7	13.3	44.8	33.7
Claude-3.5-Sonnet-1022	78.3	13.3	3.3	46.7	35.4
OpenAI-o1-preview	85.5	44.6	40.0	71.0	43.6
OpenAI-o1-mini	90.0	56.6	46.7	74.4	46.3
		7B Models			
Qwen2.5-Instrust-7B	76.6	13.3	0.0	37.0	29.1
Qwen2.5-Math-Instrust-7B	81.8	20.0	13.3	44.1	31.1
rStar-Math-7B	78.4*	26.7*	-	-	47.1*
Qwen2.5-7B-SimpleRL	82.4*	26.7*	-	-	37.6*
Eurus-2-7B-PRIME	79.2*	26.7*	-	-	42.1*
DeepSeek-R1-Distill-Qwen-7B	<u>92.8</u>*	**55.5***	**40.0**	**65.6**	<u>64.1</u>
OREAL-7B	91.0	33.3	33.3	62.6	59.9
OREAL-DSR1-Distill-Qwen-7B	**94.0**	<u>50.0</u>	**40.0**	**65.6**	**66.1**
		32B Models			
Qwen2.5-Instrust-32B	80.6	20.0	13.3	50.8	40.4
QwQ-32B-Preview	90.6	50.0	40.0	<u>72.7</u>	58.5
DeepSeek-R1-Distill-Qwen-32B	94.3*	**72.6***	**46.7**	67.7	<u>71.2</u>
OREAL-32B	**95.0**	<u>60.0</u>	**46.7**	**74.8**	**72.4**

Table 1: Overall evaluation results for OREAL and each baseline. "OREAL-DSR1-Distill-Qwen-7B" denotes the DeepSeek-R1-Distill-Qwen7B trained by OREAL. "AIME2025-I", "LiveMath" and "Olympiad" represent "AIME 2025 Part1", "LiveMathBench", and "OlympiadBench", respectively. For models at the parameter scale of 7B and 32B, we use Bold and Underlined to represent the best and second best performance, respectively. For part of the baseline, we directly use the results from their report, marked with *.

Setting	MATH-500
Initial Policy	84.8
+ REINFORCE (baseline)	85.8
+ Reward Shaping	86.6
+ Behavior Cloning	87.6
+ Token Re-weighting	89.0
+ Skill-based Enhancement	91.0

Table 2: Ablation study for 7B models performance on MATH-500 with different reinforcement learning settings.

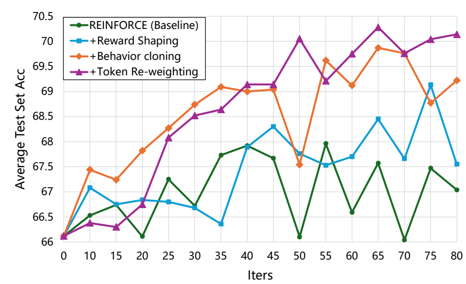

Figure 2: Average test accuracy of 7B models across different training steps.

In addition, OREAL-DSR1-Distill-Qwen-7B improves the MATH-500 score from 92.8 to 94.0 and also achieves gains on LiveMathBench and OlympiadBench. However, its performance on the AIME benchmark series is comparatively weaker. We observe the same disadvantages of OREAL-32B and OREAL-7B, whose AIME2024 scores are relatively lower than the best scores. Since the overall performance verifies the effectiveness of the OREAL algorithm, we attribute the reason to the deficiency (*e.g.*, in response quality, query difficulty, and quantity) of RFT data and RL training queries for obtaining high performance in the domain of AIME and leave it for the future work.

4.3 Ablation Study

To verify the effectiveness of each component described in Section 2, we progressively add the proposed component based on the 7B model and compare the evaluation results on MATH-500, starting from REINFORCE (Sutton et al., 1999) as baseline.

As shown in Tabel 2, we add each component step by step, where "Reward Shaping" represents L_2 introduced in Section 2.3, "Behavior Cloning" represents L_1 introduced in Section 2.2, "Token Re-weighting" represents L_{total} introduced in Section 2.4. The gradual addition of the modules steadily increases the Pass@1 scores of the 7B model on MATH-500, proving the effectiveness of our method. Ultimately, the policy model is raised from an initial score of 84.8 to 91.0.

We also report average pass@1 accuracy across all benchmarks during the training process with different RL settings. As shown in Figure 2, the REINFORCE training process is unstable, which can be mitigated by "Reward Shaping". "Behavioral Cloning" for positive samples can speed up convergence and show better performance early in training. Although the performance growth of "Token Re-weighting" is relatively slow in the early stage of training, it ultimately obtains the best results.

5 Related Work

Stimulate Reasoning using Chain of Thought. In mathematical reasoning tasks, Chain of Thought (CoT) (Wei et al., 2022) is recognized as a crucial technique to enhance the reasoning ability of large language models (LLMs), which can be implemented through few-shot examples (Wei et al., 2022) or prompt engineering (Kojima et al., 2022). Self-consistency (Wang et al., 2022) is further proposed to generate and voting through multiple CoTs. In addition to simple CoTs, various search methods have been explored that simultaneously consider multiple potential CoTs, such as Tree-of-Thought (ToT) (Yao et al., 2024) and Graph-of-Thought (GoT) (Besta et al., 2024), which extend the idea to tree or graph structure, offering more flexibility in developing CoTs and backtracking. However, these methods mainly stimulate the reasoning capability of LLMs by prompts without parameter updates, these inference-time techniques do not fundamentally improve the underlying ability of LLMs.

Reasoning Enhancement by Supervised Fine-tuning. To let the LLMs essentially acquire the reasoning abilities, many studies Ying et al. (2024); Yu et al. (2023); Liu et al. (2024b); Li et al. (2023); Liu & Low (2023); Yue et al. (2023) have explored synthesizing high-quality data to conduct supervised fine-tuning (SFT) on LLMs. But this method heavily relies on high-quality training data and a existing high-performing model (Lightman et al., 2023). As a result, many existing works (Huang et al., 2024; Min et al., 2024) have turned to distilling knowledge from powerful, large-scale models to synthesize data, yielding good results. However, distilled-based methods receive the limitations of the teacher model. Some studies argue that such approaches merely transform the model into a knowledge retriever, rather than an actual reasoner (Kambhampati, 2024).

Reinforcement Learning for LLM. Compared to SFT, reinforcement learning (RL) offers better generalization and is therefore considered a more fundamental training aproach (Chu et al., 2025). The advent of the o1 family of models (OpenAI, 2024b) and a series of o1-like works (DeepSeek-AI et al., 2025; Zeng et al., 2025; Cui et al., 2025; Guan et al., 2025) make the importance of large-scale RL for inference became more apparent. Currently, the mainstream approach to RL involves using outcome reward signals (DeepSeek-AI et al., 2025; Zeng et al., 2025; Kazemnejad et al., 2024) and there are different views in the community on how to use that reward signal. ReST^{EM} (Singh et al., 2023) and RFT (Yuan et al., 2023) simply select the positive samples based on the binary signal and only use them for behavior cloning. GRPO (Shao et al., 2024), RLOO (Fukunaga & Hummels, 1989; Ahmadian et al., 2024), REINFORCE (Sutton et al., 1999), use both positive and negative samples for policy updating, but facing the challenges of sparse reward in long sequence. PPO (Schulman et al., 2017) makes the preference modeling on sequence-level. Different from them, to explore the limit of outcome reward, OREAL presents a unified framework, grounded in the unique characteristics of mathematical reasoning tasks.

6 Conclusion

This paper aims to explore the limit of **O**utcome **RE**w**A**rd-based reinforcement **L**earning for mathematical reasoning tasks, and proposes a unified policy optimization framework, termed OREAL, grounded in three key insights: 1) Behavior cloning on positive trajectories from Best-of-n (BoN) sampling is both necessary and sufficient for optimal policy learning under binary feedback; 2) Accordingly, a reward-shaping mechanism should be further introduced to transform the reward function derived from the optimal policy; 3) An efficient token-level credit assignment scheme can be achieved through trajectory advantage decomposition without relying on additional value networks. Together, these components form a theoretically grounded, general, and scalable approach for mathematical reasoning tasks. With OREAL, we are the first to improve the performance of a 7B model on the MATH-500 accuracy to 91 using the RL method instead of distillation, which even surpasses OpenAI-o1-mini and QwQ-32B-Preview. Even when taking the previous best 7B model, DeepSeek-R1-Distill-Qwen-7B, as initial policy, OREAL can improve it to 94 accuracy on MATH-500, being even on par with the previous best 32B models. OREAL-32B also obtains new state-of-the-art results among the 32B model on MATH-500, LiveMathBench, and OlympiadBench.

Along with the experimental observations presented in this paper, we also find two factors that are crucial for the success of scalable RL for mathematical reasoning tasks, which become the primary focus of our future work. First, the initial policy model should be as free of knowledge deficiencies as possible, as this serves as the foundation for further improvement during the RL stage. A strong starting point ensures that RL can effectively and efficiently incentivize the underlying capability of LLMs obtained through pre-training or supervised fine-tuning. Towards this goal, it is a practical way to conduct distillation or data synthesis with DeepSeek-R1 or DeepSeek-V3, which is not explored in this work as it is orthogonal to our investigation. Second, the quality of the data used in the RL phase must be diverse and sufficient in terms of difficulty, quantity, and scope. A well-balanced dataset enables the model to reach its full potential by exposing it to a broad range of challenges and learning opportunities. Thus, we believe it is still valuable to make efforts in the pre-training and post-training data construction process.

Acknowledgement

We thank the anonymous reviewers and area chair for their helpful comments. This project is funded in part by the Hong Kong Generative AI Research and Development Center (HKGAI) under the Innovation and Technology Commission (ITC)'s InnoHK. Dahua Lin is a PI of HKGAI under the InnoHK. This project is also supported by the Shanghai Artificial Intelligence Laboratory.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.

Anthropic. Claude 3.5 sonnet, 2024. URL `https://www.anthropic.com/news/claude-3-5-sonnet`.

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.

Yinlam Chow, Guy Tennenholtz, Izzeddin Gur, Vincent Zhuang, Bo Dai, Sridhar Thiagarajan, Craig Boutilier, Rishabh Agarwal, Aviral Kumar, and Aleksandra Faust. Inference-aware fine-tuning for best-of-n sampling in large language models. *arXiv preprint arXiv:2412.15287*, 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. *arXiv preprint arXiv:2501.17161*, 2025.

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.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward model ensembles help mitigate overoptimization. *arXiv preprint arXiv:2310.02743*, 2023.

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

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, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Keinosuke Fukunaga and Donald M. Hummels. Leave-one-out procedures for nonparametric error estimates. *IEEE transactions on pattern analysis and machine intelligence*, 11(4): 421–423, 1989.

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

Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, and Marc Dymetman. Compositional preference models for aligning lms. *arXiv preprint arXiv:2310.13011*, 2023.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*, 2025.

Lin Gui, Cristina Gârbacea, and Victor Veitch. Bonbon alignment for large language models and the sweetness of best-of-n sampling. *arXiv preprint arXiv:2406.00832*, 2024.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024a.

Conghui He, Wei Li, Zhenjiang Jin, Chao Xu, Bin Wang, and Dahua Lin. Opendatalab: Empowering general artificial intelligence with open datasets, 2024b. URL https://arxiv.org/abs/2407.13773.

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. *arXiv preprint arXiv:2103.03874*, 2021.

Jacob Hilton and Leo Gao. Measuring goodhart's law. *OpenAI Research Blog*, 2022.

Zhen Huang, Haoyang Zou, Xuefeng Li, Yixiu Liu, Yuxiang Zheng, Ethan Chern, Shijie Xia, Yiwei Qin, Weizhe Yuan, and Pengfei Liu. O1 replication journey–part 2: Surpassing o1-preview through simple distillation, big progress or bitter lesson? *arXiv preprint arXiv:2411.16489*, 2024.

Harold Jeffreys. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 186(1007): 453–461, 1946.

Subbarao Kambhampati. Can large language models reason and plan? *Annals of the New York Academy of Sciences*, 1534(1):15–18, 2024.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Unlocking rl potential for llm reasoning through refined credit assignment. *arXiv preprint arXiv:2410.01679*, 2024.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. T\" ulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Chengpeng Li, Zheng Yuan, Guanting Dong, Keming Lu, Jiancan Wu, Chuanqi Tan, Xiang Wang, and Chang Zhou. Query and response augmentation cannot help out-of-domain math reasoning generalization. *arXiv preprint arXiv:2310.05506*, 2023.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13:9, 2024.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Haoxiong Liu, Yifan Zhang, Yifan Luo, and Andrew Chi-Chih Yao. Augmenting math word problems via iterative question composing. *arXiv preprint arXiv:2401.09003*, 2024b.

Junnan Liu, Hongwei Liu, Linchen Xiao, Ziyi Wang, Kuikun Liu, Songyang Gao, Wenwei Zhang, Songyang Zhang, and Kai Chen. Are your llms capable of stable reasoning? *arXiv preprint arXiv:2412.13147*, 2024c.

Tiedong Liu and Bryan Kian Hsiang Low. Goat: Fine-tuned llama outperforms gpt-4 on arithmetic tasks. *arXiv preprint arXiv:2305.14201*, 2023.

Wentao Liu, Hanglei Hu, Jie Zhou, Yuyang Ding, Junsong Li, Jiayi Zeng, Mengliang He, Qin Chen, Bo Jiang, Aimin Zhou, et al. Mathematical language models: A survey. *arXiv preprint arXiv:2312.07622*, 2023.

MAA. American invitational mathematics examination - aime. In *American Invitational Mathematics Examination - AIME*. URL https://maa.org/math-competitions/american-invitational-mathematics-examination-aime.

Nikolaos Matzakos, Spyridon Doukakis, and Maria Moundridou. Learning mathematics with large language models: A comparative study with computer algebra systems and other tools. *International Journal of Emerging Technologies in Learning (iJET)*, 18(20):51–71, 2023.

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

OpenAI. Hello GPT-4o, 2024a. URL https://openai.com/index/hello-gpt-4o/.

OpenAI. Learning to reason with llms. 2024b.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. Training language models with language feedback at scale. *arXiv preprint arXiv:2303.16755*, 2023.

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

Pier Giuseppe Sessa, Robert Dadashi, Léonard Hussenot, Johan Ferret, Nino Vieillard, Alexandre Ramé, Bobak Shariari, Sarah Perrin, Abe Friesen, Geoffrey Cideron, et al. Bond: Aligning llms with best-of-n distillation. *arXiv preprint arXiv:2407.14622*, 2024.

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

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585*, 2023.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

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

Qwen Team. Qwq: Reflect deeply on the boundaries of the unknown, November 2024. URL https://qwenlm.github.io/blog/qwq-32b-preview/.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, et al. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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. *Advances in Neural Information Processing Systems*, 36, 2024.

Huaiyuan Ying, Shuo Zhang, Linyang Li, Zhejian Zhou, Yunfan Shao, Zhaoye Fei, Yichuan Ma, Jiawei Hong, Kuikun Liu, Ziyi Wang, et al. Internlm-math: Open math large language models toward verifiable reasoning. *arXiv preprint arXiv:2402.06332*, 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. *arXiv preprint arXiv:2309.12284*, 2023.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*, 2023.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

Weihao Zeng, Yuzhen Huang, Wei Liu, Keqing He, Qian Liu, Zejun Ma, and Junxian He. 7b model and 8k examples: Emerging reasoning with reinforcement learning is both effective and efficient. `https://hkust-nlp.notion.site/simplerl-reason`, 2025. Notion Blog.

Tianyang Zhong, Zhengliang Liu, Yi Pan, Yutong Zhang, Yifan Zhou, Shizhe Liang, Zihao Wu, Yanjun Lyu, Peng Shu, Xiaowei Yu, et al. Evaluation of openai o1: Opportunities and challenges of agi. *arXiv preprint arXiv:2409.18486*, 2024.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A Binary Feedback under Outcome Supervision.

Though a reasoning trajectory usually contains multiple reasoning steps with thousands of tokens, there lacks an efficient approach to automatically label the correctness of each token or reasoning step in math reasoning tasks. Thus, a practical way is to parse the final answer from the reasoning trajectory (DeepSeek-AI et al., 2025; Lambert et al., 2024), evaluate its correctness based on rules or models, and then provide an outcome reward at the end of the trajectory as below:

$$R(s_t) = \begin{cases} 1 & \text{if } t \text{ is the end step and the answer is correct} \\ 0 & \text{otherwise,} \end{cases} \tag{A1}$$

which intrinsically treats the correct trajectory equally for learning. Moreover, the reward signal is severely sparse when compared to the thousands of tokens and does not provide any signal of progress or correctness for intermediate steps. The resulting reward distribution of trajectories is also different from that of the dense reward function constructed through preference pairs in traditional RL for large language models (Ouyang et al., 2022), which induces a more appropriate optimization framework for mathematical reasoning tasks, discussed in the next section.

B Hyperparameters Details

During training iterations, each batch consists of 64 questions, with 16 rollouts per question. The max length of each rollout trajectory is set to 16384 tokens. Then the correctness of each response is averaged to calculate the pass rate, and questions with an overall pass rate of 0 or 1 are discarded. For the remaining trajectories, we retain only one correct response and one incorrect response per question, ensuring a balanced distribution of positive and negative samples for token-level reward model training.

For optimization, the policy model is trained with a learning rate of $5e{-}7$, while the token-level reward model is trained with a learning rate of $2e{-}6$. The latter undergoes a 10-step warm-up phase before training begins. Both models employ a cosine annealing learning rate schedule, decaying to $1/5$ of the initial learning rate over time. We optimize both models using the AdamW optimizer. The total number of training steps is 80, with evaluation conducted every 10 steps. The KL coefficient β is set to 0.01. We select the best-performing model determined by evaluation metrics.

C Analysis of the Impact of Initial Policy Models

C.1 Skill-based Enhancement

During the RL training procedure, we observe that the model consistently struggles with certain types of questions, particularly those involving specific knowledge and skill areas, such as trigonometric constant transformations, probability statistics, series transformations, *etc*. We believe this is caused by the insufficient learning of the base model on these concepts in the Pre-training or RFT stages.

To address this problem, we implement a skill-based enhancement approach, using the MATH dataset to reduce the high cost of skill annotation. Specifically, we annotate each question in the training set with its corresponding core skill. For questions that the model repeatedly fails to answer correctly during the RL phase, we perform data augmentation by including similar questions from the training set that share the same skill. These augmented questions are then added to the training data during the RFT stage to help the model better internalize these skills.

C.2 Results

We further analyze OREAL by adopting it to several different initial policy models, as shown in Table A1. OREAL consistently improves the performance of each initial policy

Model	MATH-500	AIME2024	AIME2025-I	LiveMath	Olympiad
OREAL-7B-SFT-wo-enhance	84.8	26.7	26.7	55.0	55.1
OREAL-7B-wo-enhance	89.0	36.7	40.0	60.1	58.1
OREAL-7B-SFT	86.4	26.7	26.7	54.2	56.0
OREAL-7B	91.0	33.3	33.3	62.6	59.9
DeepSeek-R1-Distill-Qwen-7B	92.8*	55.5*	40.0	65.6	64.1
OREAL-DSR1-Distill-Qwen-7B	94.0	50.0	40.0	65.6	66.1
OREAL-32B-SFT	92.6	43.3	46.7	71.9	68.7
OREAL-32B	95.0	60.0	46.7	74.8	72.4

Table A1: Evaluation for the performance of OREAL on different initial policy models. Here, "-SFT" and "DeepSeek-R1-Distill-Qwen7B" denote the initial policy model. "wo-enhance" means the model which do not perform the skill-based enhancement during the SFT stage.

model, including our own trained model and the strong distilled model (DeepSeek-AI et al., 2025), on MATH-500, LiveMathBench, and OlympiadBench, except slight fluctuations on AIME2024 and AIME2025 part1 when the performance of initial policy models are already high (*e.g.*, DeepSeek-R1-Distill-Qwen-7B), which demonstrates the generality of OREAL.

After adding skill-based enhancement data, there is a significant rise in MATH-500 scores for the initial policy model (row 1 and row 3) and the corresponding RL-trained model (row 2 and row 4). Since our enhancement is performed primarily for the MATH-500, this verifies the effectiveness of the skill-based enhancement approach. In addition, the performance of the model after RL is strongly correlated with the capabilities of the initial policy model itself. The stronger the initial policy model, the higher the performance that RL can deliver, indicating the importance of policy initialization.

D Token Level Reward Model

D.1 Training Settings

For the token-level reward model, we directly use the binary outcome rewards provided by the verifier and optimize using the cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = -\mathbb{E}_{(s,r)\sim\mathcal{D}}\left[r \log p(s) + (1-r)\log(1-p(s))\right], \tag{A2}$$

where s represents the sampled trajectory, $r \in \{0, 1\}$ is the binary outcome reward from the verifier, and $p(s) = \sigma(\frac{1}{T}\sum_t^T w(s_t))$ denotes the predicted probability of correctness by the token-level reward model w.

To further analyze the behavior of the token-level reward model, we visualize its output distribution $w(s_t)$ during the on-policy RL training process (see Appendix D.2). In this training paradigm, $w(s_t)$ assigns token-wise importance scores across the chain-of-thought reasoning process, capturing each token's contribution to the final correctness of the generated response. Consequently, this allows us to leverage $w(s_t)$ for importance sampling during the optimization process, enabling a more principled selection of informative tokens.

D.2 Score Visualization

Figure A1 and A2 show the token-level reward model scores across responses. The values are normalized to [0, 1]. Cooler colors indicate higher reward scores, while warmer colors denote lower scores. For correct responses, the overall REWARDS are high, especially at the end, although there are a few lower sections in the middle. For incorrect responses, the distribution of rewards is reversed, and the closer to the end the lower the rewards. This indicates that not all tokens contribute to the response equally and it is important to assign token-level credits to the sequences.

Figure A1: Token-level reward model score visualization for a correct response.

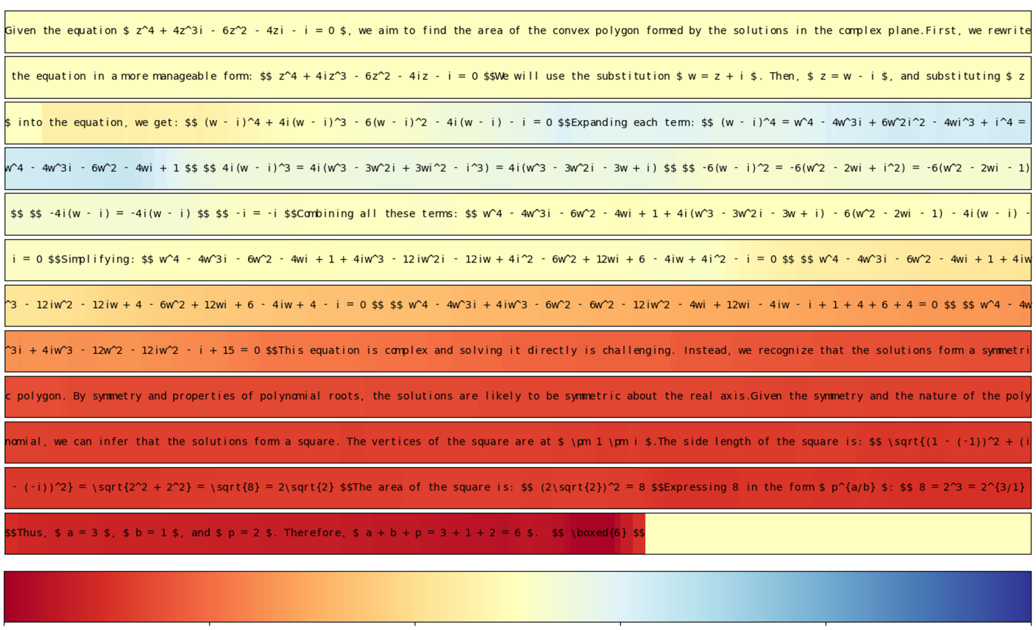

Figure A2: Token-level reward model score visualization for an incorrect response.

E Additional Discussion

E.1 Discussion about the Computational Cost of BoN Sampling

The computational cost of BoN sampling primarily stems from the varying number of sampling attempts required to obtain correct samples, which is influenced by differences in problem difficulty and model capability. To analyze the computational cost during training, we sampled the training set and divided the problems into two categories—easy and hard—based on the initial policy model's pass rate on the training set. We then evaluated the Pass@N metric of the model at different training iterations on these two subsets. Due to resource constraints, we sampled 8 responses for each query.

As shown in the Table A2, the overall scores for the easy problem sets are significantly higher than those for the hard sets, indicating that greater effort is required to sample correct answers for difficult problems. As training progressed, the model's capability steadily improves, with the pass@1 scores increasing across both easy and hard sets. This suggests a higher probability of sampling correct answers, implying that using a stronger model can theoretically reduce computational overhead. Interestingly, the Pass@8 score does not show a noticeable improvement during the training process, which is consistent with observations of Yue et al. (2025).

step	10	20	30	40	50	60
Easy/Pass@1	73.49	77.17	77.17	77.0	78.56	78.89
Easy/Pass@4	97.22	96.32	97.22	96.73	97.22	96.24
Easy/Pass@8	98.45	98.61	98.85	98.61	98.53	98.20
Hard/Pass@1	31.41	38.68	40.29	40.05	40.19	41.43
Hard/Pass@4	72.57	71.33	79.70	75.99	78.19	76.82
Hard/Pass@8	87.38	86.13	87.67	86.69	87.52	87.93

Table A2: The evaluation results for OREAL on the easy and difficult subsets at different training iterations, using Pass@N metric.

E.2 Discussion about the Generalization of OREAL

Although this paper specifically focuses on mathematical reasoning tasks, we believe that OREAL is broadly applicable to any task that provides a clear outcome reward signal (e.g., instruction following, programming problem solving, puzzle solving, etc.), because we only assume the reward is a binary signal, without making any assumptions about the specific task type.

To verify the effectiveness of our method in other domains, we conducted training and evaluation on the instruction-following task using the IFEval dataset (Zhou et al., 2023). As shown in Figure A3, the model's ability to follow instructions gradually improved during training, indicating that our method possesses generalization capability and cross-domain application potential.

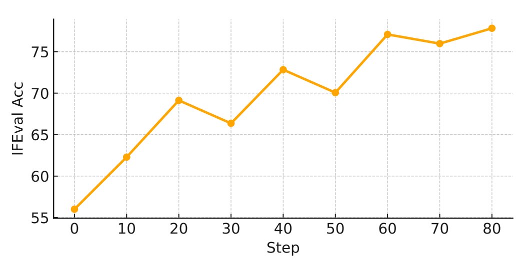

Figure A3: IFEval accuracy across different training steps.

F Prompt

Figure A4 is the system prompt of the verifier model, which is used during RL training to provide the binary outcome reward for a response. Figure A5 is the system prompt we use for fine-tuning and RL training as well as the evaluation.

Verifier Prompt:
You are a helpful assistant who evaluates the correctness and quality of models' outputs.

Please as a grading expert, judge whether the final answers given by the candidates below are consistent with the standard answers, that is, whether the candidates answered correctly.

Here are some evaluation criteria:

1. Please refer to the given standard answer. You don't need to re-generate the answer to the question because the standard answer has been given. You only need to judge whether the candidate's answer is consistent with the standard answer according to the form of the question. Don't try to answer the original question. You can assume that the standard answer is definitely correct.
2. Because the candidate's answer may be different from the standard answer in the form of expression, before making a judgment, please understand the question and the standard answer first, and then judge whether the candidate's answer is correct, but be careful not to try to answer the original question.
3. Some answers may contain multiple items, such as multiple-choice questions, multiple-select questions, fill-in-the-blank questions, etc. As long as the answer is the same as the standard answer, it is enough. For multiple-select questions and multiple-blank fill-in-the-blank questions, the candidate needs to answer all the corresponding options or blanks correctly to be considered correct.
4. Some answers may be expressed in different ways, such as some answers may be a mathematical expression, some answers may be a textual description, as long as the meaning expressed is the same. And some formulas are expressed in different ways, but they are equivalent and correct.
5. If the prediction is given with \boxed{}, please ignore the \boxed{} and only judge whether the candidate's answer is consistent with the standard answer.

Please judge whether the following answers are consistent with the standard answer based on the above criteria. Grade the predicted answer of this new question as one of:
A: CORRECT
B: INCORRECT
Just return the letters "A" or "B", with no text around it.

Here is your task. Simply reply with either CORRECT, INCORRECT. Don't apologize or correct yourself if there was a mistake; we are just trying to grade the answer.
<Original Question Begin>:
ORIGINAL QUESTION
<Original Question End>

<Gold Target Begin>:
GOLD ANSWER
<Gold Target End>

<Predicted Answer Begin>:
ANSWER
<Predicted End>

Judging the correctness of candidates' answers:

Figure A4: Prompts for the model-based generative verifier.

System Prompt:
You are an expert mathematician with extensive experience in mathematical competitions. You approach problems through systematic thinking and rigorous reasoning. When solving problems, follow these thought processes:

Deep Understanding
Take time to fully comprehend the problem before attempting a solution. Consider:
- What is the real question being asked?
- What are the given conditions and what do they tell us?
- Are there any special restrictions or assumptions?
- Which information is crucial and which is supplementary?

Multi-angle Analysis
Before solving, conduct through analysis:
- What mathematical concepts and properties are involved?
- Can you recall similar classic problems or solution methods?
- Would diagrams or tables help visualize the problem?
- Are there special cases that need separate consideration?

Systematic Thinking
Plan your solution path:
- Propose multiple possible approaches
- Analyze the feasibility and merits of each method
- Choose the most appropriate method and explain why
- Break complex problems into smaller, manageable steps

Rigorous Proof
During the solution process:
- Provide solid justification for each step
- Include detailed proofs for key conclusions
- Pay attention to logical connections
- Be vigilant about potential oversights

Repeated Verification
After completing your solution:
- Verify your results satisfy all conditions
- Check for overlooked special cases
- Consider if the solution can be optimized or simplified
- Review your reasoning process

Remember:
1. Take time to think thoroughly rather than rushing to an answer
2. Rigorously prove each key conclusion
3. Keep an open mind and try different approaches
4. Summarize valuable problem-solving methods
5. Maintain healthy skepticism and verify multiple times

Your response should reflect deep mathematical understanding and precise logical thinking, making your solution path and reasoning clear to others. When you're ready, present your complete solution with:
- Clear problem understanding
- Detailed solution process
- Key insights
- Thorough verification

Focus on clear, logical progression of ideas and thorough explanation of your mathematical reasoning. Provide answers in the same language as the user asking the question, repeat the final answer using a '\boxed{}' without any units, you have [[8192]] tokens to complete the answer.

Figure A5: System prompts for long CoT reasoning.