Early Preview Hierarchical GRPO To Boost Reasoning Of Small-Sized Large Language Models

Anonymous ACL submission

Abstract

Inference scaling enhances the reasoning capabilities of large language models, with reinforcement learning serving as the key technique to draw out complex reasoning. However, key technical details of state-of-the-art reasoning LLMs—such as those in the OpenAI O series, Claude 3 series, DeepMind's Gemini 2.5 series, and Grok 3 series—remain undisclosed, making it difficult for the research community to replicate their reinforcement learning training results. We propose an Early Preview Hierarchical Reinforcement Learning algorithm based on the open-sourced Group Relative Policy Optimization (GRPO) framework. In details, we introduce an early preview version of a hierarchical reinforcement learning approach that continues to enhance the reasoning capabilities of small-sized large language models. In particular, a 1.5B-parameter LLM achieves 53.3% on AIME and 90.4% on Math500, These results, enabled by the proposed early preview efficient hierarchical reinforcement learning, demonstrate math reasoning capabilities comparable to O1-mini/O3-mini—achievable within a typical school laboratory setting. In addition, we open-source both the dataset and model checkpoints to support future research in large-scale reinforcement learning for LLMs.

1 Introduction

Large language models (LLMs) with advanced reasoning capabilities, such as OpenAI o-series (Jaech et al., 2024; OpenAI, 2024, 2025a,b), DeepSeek R1 (Guo et al., 2025), and Claude 3.7 (Anthropic, 2025), grok3-reasoning (XAI, 2024), Gemini 2.5 (LLC, 2025), have achieved remarkable performance in complex tasks like mathematical reasoning and code generation. Through large-scale reinforcement learning (RL), these models acquire advanced reasoning strategies—such as step-by-step analysis (Wei et al., 2022), self-reflection (Wang et al., 2023), and backtracking (Ahmadian et al., 2024)—which enhance their ability to solve complex reasoning problems with greater robustness and accuracy across diverse domains.

Currently, most successful reinforcement learning efforts—including open-source research—depend on relatively large language base models, especially when aiming to improve math and code reasoning capabilities. Moreover, it has been widely believed that improving both mathematical and coding capabilities in small models is particularly challenging. To further explore the potential of reinforcement learning in enhancing reasoning abilities, we investigate the effectiveness of hierarchical reinforcement learning–trained reasoning models based on hierarchical reinforcement learning (Guo et al., 2025; Christiano et al., 2017; Sutton and Barto, 2018; Everitt et al., 2017, 2021; Weng, 2024), which shows promising potential for scalability.

In this work, we present the Early Preview Hierarchical GRPO algorithm—an early version of a hierarchical reinforcement learning method designed to improve reasoning tasks in our series of small to medium-sized large language models. Our experiments demonstrate that the proposed Early Preview Hierarchical Reinforcement Learning algorithm exhibits exceptional reasoning capabilities, outperforming many larger state-of-the-art closed-source and open-source reasoning large language models (OpenAI, 2024; Jaech et al., 2024). In detail, it demonstrates superior performance on both mathematics and code reasoning tasks, surpassing OpenAI's O1-mini, O1, and O3-mini (low) models (OpenAI, 2024; Jaech et al., 2024) within 1.5B- and 14B-parameter LLMs trained using the early preview version of the hierarchical GRPO algorithm on major reasoning benchmarks for math and coding.

2 Related Work

2.1 Reasoning Large Language Models

In the context of LLMs, reinforcement learning has been widely used for aligning human preferences (Christiano et al., 2017; Ouyang et al., 2022; Yuan et al., 2024a; Azar et al., 2024; Rafailov et al., 2023; Yuan et al., 2024a), but the open-source community mostly adopt the data-driven imitation learning methods (Yuan et al., 2024b; Yue et al., 2023; Guan et al., 2025) to enhance the reasoning capabities of LLMs. Over the past few months, the paradigm gradually shifted. OpenAI o1 (Jaech et al., 2024) first showed the tremendous potential of large-sacle RL for reasoning LLMs, and recent works have verified the scaling effect of the simple RL recipe with merely outcome rewards (Guo et al., 2025; Qwen Team, 2024; XAI, 2024). Meanwhile, the role of dense rewards in RL remains underexplored, which is the main focus of PRIME (Cui et al., 2025). Unfortunately, only outcome reward models (ORMs) (Guo et al., 2025) are available in most practices of LLMs, i.e., only the final token bears a meaningful reward while intermediate tokens receive no rewards (Rafailov et al., 2023; Shao et al., 2024; Guo et al., 2025). Very recently, the state-of-the-art reasoning models OpenAI o-series (Jaech et al., 2024; OpenAI, 2024, 2025a,b), DeepSeek R1 (Guo et al., 2025), and Claude 3.7 (Anthropic, 2025), grok3-reasoning (XAI, 2024), Gemini 2.5 (LLC, 2025), have achieved remarkable performance in complex tasks like mathematical reasoning and code generation. However, deep reinforcement learning algorithm is not well explored on the reasoning ability of small-size (0.7B/1.5B) large language models with support of small scale of math dataset and school-lab resource.

2.2 Reinforcement Learning To Enhance LLM Reasoning

Reinforcement learning (RL) has demonstrated strong potential in enhancing the reasoning abilities of LLMs across various domains, including mathematics (Guo et al., 2025; Jaech et al., 2024) and coding (OpenAI, 2025b; LLC, 2025). Long-chain-of-thought (long-COT) LLMs, such as OpenAI-O3 (OpenAI, 2025a) and DeepSeek-R1 (Guo et al., 2025), significantly outperform their short-COT counterparts. These models demonstrate that reinforcement learning with verifiable rewards (RLVR) can effectively promote deep reasoning behaviors—such as broad exploration and feasibility checks (Gandhi et al., 2025)—without the need for complex reasoning data generation techniques like Monte Carlo Tree Search (Hosseini et al., 2024; Yang et al., 2024). However, these behaviors often result in significantly longer reasoning traces—sometimes several times longer than those generated by short-COT LLMs (Wang et al., 2024; Zhang et al., 2024b)—leading to an 'overthinking' problem that substantially increases inference costs (Kumar et al., 2025). Recent studies have shown that extended reasoning often includes redundant or unnecessary verification and reflection, even on simple problems (Shao et al., 2024; KimiTeam et al., 2025). Other studies, such as (Hao et al., 2024; Geiping et al., 2025), represent reasoning as an optimization over latent vectors rather than text tokens, enabling a more efficient and concise reasoning process. To reduce the reasoning length of trained LLMs, several test-time methods—such as early-exit strategies—have been developed (Muennighoff et al., 2025; Fu et al., 2024; Zhang et al., 2024a). However, hierarchical reinforcement learning is not well studied to boost the reasoning ability of small-sized large language models with support of small scale of math dataset.

2.3 Hierarchical Reinforcement Learning

Hierarchical Reinforcement Learning (HRL) (Sutton and Barto, 2018) offers the advantages of temporal abstraction and enhanced exploration efficiency (Nachum et al., 2018). The options architecture (Sutton and Barto, 2018; Bacon et al., 2017; Harutyunyan et al., 2018; Klissarov et al., 2017; Kaelbling, 1993; Gao et al., 2024; Dayan and Hinton, 1993a; Salter et al., 2022b) learns temporally extended macro-actions along with a termination function, offering an elegant framework for hierarchical reinforcement learning. In goal-conditioned feudal learning (Dayan and Hinton, 1993b; Vezhnevets et al., 2017), a higher-level agent generates subgoals for a lower-level agent, which then executes atomic actions in the environment. To address the resulting non-stationarity, prior works (Nachum et al., 2018; Levy et al., 2018) propose relabeling previously collected transitions to train goal-conditioned policies more effectively. Prior methods (Rajeswaran et al., 2018; Nair et al., 2018; Hester et al., 2018; Shiarlis et al., 2018; Fox et al., 2017; Kipf et al., 2019; Zhang et al., 2020; Pertsch et al., 2020; Chane-Sane et al., 2021; Kreidieh et al., 2020; Singh et al., 2021) leverage expert demonstrations to improve sample efficiency and accelerate learning, particularly for task segmentation. Other approaches either utilize bottleneck option discovery (Salter et al., 2022a) or behavior priors (Salter et al., 2022b) to identify and embed behaviors from past experience, or rely on hand-designed action primitives (Dalal et al., 2021; Nasiriany et al., 2022). Inspired by the potential of hierarchical reinforcement learning, we study the effectiveness of hierarchal reinforcement learning to boost the math reasonning ability of small-sized large language models.

3 Method

3.1 Preliminary: LLM Reasoning Via GRPO+ (Yu et al., 2025)

3.1.1 Group Relative Policy Optimization(Shao et al., 2024)

Compared to Proximal Policy Optimization (PPO) (Schulman et al., 2017), Group-Relative Policy Optimization (GRPO) (Shao et al., 2024) eliminates the value function and estimates the advantage in a group-relative manner.

For a specific question-answer pair (q, a), the behavior policy $\pi_{\theta_{\text{old}}}$ samples a group of G individual responses $\{o_i\}_{i=1}^{G}$. Then, the advantage of the i-th response is calculated by normalizing the group-level rewards $\{R_i\}_{i=1}^{G}$ as follows:

$$\hat{A}_{i,t} = \frac{r_i - \text{mean}(\{R_i\}_{i=1}^{G})}{\text{std}(\{R_i\}_{i=1}^{G})}. \tag{1}$$

Similar to PPO, GRPO adopts a clipped objective, together with a directly imposed KL penalty term:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta\text{old}}(\cdot|q)} \left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left(\min\left(r_{i,t}(\theta)\hat{A}_{i,t},\ \text{clip}\left(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon\right)\hat{A}_{i,t}\right) - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\right)\right] \tag{2}$$

where

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi\_\theta_{\text{old}}(o_{i,t} \mid q, o_{i,<t})}. \tag{3}$$

It is also worth noting that GRPO (Shao et al., 2024) computes the objective at the sample-level. To be exact, GRPO first calculates the mean loss within each generated sequence, before averaging the loss of different samples. As we will be discussing in Section 3.3, such difference may have an impact on the performance of the algorithm. where μ_R and σ_R are the mean and standard deviation of the rewards in the group:

3.1.2 Group Relative Policy Optimization Plus(GRPO+)

The advanced Group Relative Policy Optimization algorithm (Yu et al., 2025) is then developed. It samples a group of outputs $\{o_i\}_{i=1}^{G}$ for each $question q$ paired with the answer a, and optimizes the policy via the following objective:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim D_q,\,\{o_i\}_{i=1}^{G}\sim\pi_\theta(\cdot|q)} \left[\frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{j=1}^{|o_i|}\min\left(\frac{\pi_\theta(o_i \mid q)}{\pi_{\theta_{\text{old}}}(o_i \mid q)}A_{i,j},\ \text{clip}\left(\frac{\pi_\theta(o_i \mid q)}{\pi_{\theta_{\text{old}}}(o_i \mid q)},\ 1-\varepsilon_{\text{low}},\ 1+\varepsilon_{\text{high}}\right)A_{i,j}\right)\right] \tag{4}$$

where

$$A_{i,j} = \frac{r_i - \text{mean}(\{r_i\}_{i=1}^{G})}{\text{std}(\{r_i\}_{i=1}^{G})} \tag{5}$$

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi\_\theta_{\text{old}}(o_{i,t} \mid q, o_{i,<t})}. \tag{6}$$

Then, the key enhancements are represented as the following:

3.1.3 Enhancements

Removal of KL Loss (Kullback and Leibler, 1951) The KL penalty term $\beta D_{\text{KL}}(\pi_\theta \parallel \pi_{\text{ref}})$ is used to control the divergence between the learned (online) policy and a fixed reference policy, thereby encouraging stable and conservative policy updates. However, during training of the long-CoT reasoning model, the policy distribution can diverge substantially from the initial model, making the KL constraint less relevant. As a result, we omit the KL penalty term from our proposed algorithm.

Clip (Schulman et al., 2017)-Higher We observed an entropy collapse phenomenon, where the policy's entropy rapidly decreases as training progresses. As a result, the sampled responses within certain groups become nearly identical. This behavior suggests limited exploration and premature convergence to a deterministic policy, which can impede effective scaling. To mitigate this issue, we propose the *Clip-Higher* strategy. Clipping

Algorithm 1 HGRPO Level-Wise Rollout

```
Input: skills $\pi_{\theta_{l-1}}(a \mid s, z)$, manager $\pi_{\theta_l}(z \mid s)$, time-commitment bounds $P^l_{\min}$ and $P^l_{\max}$, horizon $H^l$, rollout pass threshold $pass^{l-1}$, reward $r^{l-1}$.
Reset environment: $s^l_0 \sim \rho^l_0$, $t \leftarrow 0$
while $t < H^l$ do
    Sample time-commitment $p^l \sim$ Cat$([P^l_{\min}, P^l_{\max}])$
    Sample skill $a^{l-1}_t \sim \pi_{\theta_l}(\cdot \mid s^{l-1}_t)$
    if $r^{l-1}(a^{l-1}_t) > pass^{l-1}$ then
        for $t' = t$ to $t + p^l - 1$ do
            Sample action $a^l_{t'} \sim \pi_{\theta_l}(\cdot \mid s^l_{t'})$
            Observe new state $s^l_{t'+1}$ and reward $r^l_{t'}$
        end for
    else
        Continue updating gradient at level $l - 1$
    end if
    $t \leftarrow t + p^l$
end while
Output: $(s^l_0, a^{l-1}_0, a^l_0, s^l_1, a^l_1, \ldots, s^l_H, a^{l-1}_H, a^l_H, s^l_{H+1})$
```

the importance sampling ratio, as introduced in Clipped Proximal Policy Optimization (PPO-Clip) (Schulman et al., 2017), serves to constrain the trust region and improve the stability of reinforcement learning. We observe that the upper clipping threshold can limit the policy's ability to explore. Specifically, it is often easier to increase the probability of a likely *exploitation token* than to boost the probability of a less likely *exploration token*, due to the constraints imposed by ε_{low} and $\varepsilon_{\text{high}}$.

Dynamic Sampling s.t. $0 < |\{o_i \mid \text{is\_equivalent}(a, o_i)\}| < G$., To this end, we propose over-sampling and filtering out prompts with accuracy values of 1 or 0, ensuring that all remaining prompts in the batch contribute effective gradients while maintaining a consistent batch size. Before training, we continuously sample until the batch contains only examples with accuracy strictly between 0 and 1.

Token-Level Policy Gradient Loss (KimiTeam et al., 2025) To overcome the aforementioned limitations in the long-CoT RL setting, we introduce a Token-level Policy Gradient Loss that assigns greater weight to longer sequences, allowing them to have a stronger impact on the overall gradient update compared to shorter sequences. Furthermore, from the perspective of individual tokens, any generation pattern that leads to an increase or decrease in reward is reinforced or suppressed equally, regardless of the length of the response in which it appears.

3.2 Early Preview Hierarchical GRPO

We define a discrete-time finite-horizon discounted Markov decision process (MDP) by a tuple $M = (S, A, \mathcal{P}, r, \rho_0, \gamma, H)$, where S is a state set, A is an action set, $\mathcal{P} : S \times A \times S \rightarrow \mathbb{R}_+$ is the transition probability distribution, $\gamma \in [0, 1]$ is a discount factor, and H the horizon. Our objective is to find a stochastic policy π_θ that maximizes the expected discounted return within the MDP,$\eta(\pi_\theta) = \mathbb{E}_\tau \left[\sum_{t=0}^{H} \gamma^t r(s_t, a_t)\right]$. We use $\tau = (s_0, a_0, \ldots)$ to denote the entire state-action trajectory, where $s_0 \sim \rho_0(s_0), a_t \sim \pi_\theta(a_t|s_t), s_{t+1} \sim \mathcal{P}(s_{t+1}|s_t, a_t)$.

In this work, we propose a method to learn a hierarchical policy and efficiently adapt all the levels in the hierarchy to perform a new task. We study hierarchical policies composed of a higher level, or manager $\pi_{\theta_{high}}(a_{t_{high}}|s_{t_{high}})$, and a lower level, or sub-policy $\pi_{\theta_{low}}(a_{t_{low}}|s_{t_{low}})$. The higher level does not take actions in the environment directly, but rather outputs a command. The manager typically operates at a lower frequency than the sub-policies, only observing the environment every p time-steps. When the manager receives a new observation, it decides which low level policy to commit to for p environment steps. To be noted, the hierarchy contains L levels, $high = l+1$, $low = l$,

Algorithm 2 Early Preview HGRPO1: Early Preview Hierarchical GRPO1 Difficulty Order Extension Optimization

Require: initial policy model π_θ; reward model $\{R^l\}$; task prompts $\{\mathcal{D}^l\}$ with corresponding difficulty level $\{\mathcal{Q}^l\}$; hyperparameters $\{\varepsilon^l_{\text{low}}\}, \{\varepsilon^l_{\text{high}}\}$, $l = 1, 2, \ldots, L$, $\boldsymbol{Q^{l-1}} \le \boldsymbol{Q^l}$. Length Reward $\{\mathcal{K}^l\}$ with corresponding max length $\{\boldsymbol{Len^l_{\max}}\}$, $\boldsymbol{Len^{l-1}_{\max}} \le \boldsymbol{Len^l_{\max}}$.
Ensure: π_θ

```
for l = 1, ..., L do
    for step^l = 1, ..., H^l do
        Sample a batch D^l_b from D^l
        Update the old policy model π_θold ← π_θ
        Sample G^l outputs {o^l_i}_{i^l=1}^{G^l} ~ π_θold(· | q^l) for each question q^l ∈ D^l_b
        Compute rewards {r^l_{i^l}}_{i^l=1}^{G^l} for each sampled output o^l_i by running R^l
        Filter out o^l_i and add the remaining to the dynamic sampling buffer (Dynamic Sampling Equation (11))
        if buffer size n^l_b < N^l then
            continue
        end if
        For each o^l_i in the buffer, compute Â^l_{i^l,t^l} for the t^l-th token of o^l_i (Equation (9))
    end for
    for iteration = 1, ..., μ^l do
        Update the policy model π_θ by maximizing the GRPO+ objective combining with Length Reward K^l
    end for
end for
```

where $l = \{0, 1, .., L-1\}$.

3.2.1 Reformulation

Then, we propose the early preview hierarchical grpo algorithm(HGRPO). HGRPO samples a group of outputs $\{o^l_i\}_{i=1}^{G^l}$ for each question q^l_i paired with the answer a^l, $l = \{1, ..., L\}$ and optimizes the policy via the following objective:

$$\mathcal{J}_{\text{GRPO}^{Her}}(\theta) = \prod_{l=1}^{L} \mathbb{E}_{q^l \sim D^l_q,\, \{o^l_i\}_{i=1}^{G^l} \sim \pi_\theta(\cdot|q^l)}$$
$$\prod_{l=1}^{L} \left[\frac{1}{\sum_{i^l=1}^{G^l} |o^l_i|} \sum_{i^l=1}^{G^l} \sum_{j^l=1}^{|o^l_i|} \min\left(\frac{\pi_\theta(o^l_i \mid q^l)}{\pi_{\theta_{\text{old}}}(o^l_i \mid q^l)} A^l_{i^l,j^l}, \right.\right.$$
$$\left.\left. \text{clip}\left(\frac{\pi_\theta(o^l_i \mid q^l)}{\pi_{\theta_{\text{old}}}(o^l_i \mid q^l)}, 1 - \varepsilon^l_{\text{low}}, 1 + \varepsilon^l_{\text{high}} \right) A^l_{i^l,j^l} \right) \right] \tag{7}$$

where L is the total number of levels in the Early Preview GRPO Hierarchy. Similarly, l denotes the index of lever in the L hierarchy.

3.2.2 Early Preview HGRPO Level-Wise Rollout

Most hierarchical methods either consider a fixed time-commitment to the lower level skills (Florensa et al., 2017a; Frans et al., 2018), or implement the complex options framework (Precup, 2000; Bacon et al., 2017). In this work we propose an in-between, where the time-commitment to the skills is a random variable sampled from a fixed distribution Categorical(Tmin , Tmax) just before the manager takes a decision. This modification does not hinder final performance, and we show it improves zero-shot adaptation to a new task. This approach to sampling rollouts is detailed in Algorithm 1.

3.2.3 Implementation

The, we implement our proposed preview hierarchical grpo+(HGRPO) algorithm with two versions for the reinforcement learning training of reasoning LLMs. Particularly, the implementation is made to take the difficulty of the reasoning tasks in accordance with the hierarchy in the proposed preview HGRPO. The details of the proposed two imple-

Algorithm 3 Early Preview HGRPO2: Early Preview Hierarchical GRPO2 Difficulty Re-Order Extension Optimization

Require: initial policy model π_θ; reward model $\{R^l\}$; task prompts $\{\mathcal{D}^l\}$ with corresponding difficulty level $\{\mathcal{Q}^l\}$; hyperparameters $\{\varepsilon^l_{\text{low}}\}, \{\varepsilon^l_{\text{high}}\}$, $l = 1, 2, \ldots, L$, $\boldsymbol{Q^{l-1} > Q^l}$. Length Reward $\{\mathcal{K}^l\}$ with corresponding max length $\{\boldsymbol{Len^l_{\max}}\}$, $\boldsymbol{Len^{l-1}_{\max} = Len^l_{\max}}$.
Ensure: π_θ

```
for l = 1, ..., L do
    for step = 1, ..., M do
        Sample a batch D^l_b from D^l
        Update the old policy model π_θold ← π_θ
        Sample G^l outputs {o^l_i}_{i^l=1}^{G^l} ~ π_θold(· | q^l) for each question q^l ∈ D^l_b
        Compute rewards {r^l_{i^l}}_{i^l=1}^{G^l} for each sampled output o^l_i by running R^l
        Filter out o^l_i and add the remaining to the dynamic sampling buffer (Dynamic Sampling Equation (11))
        if buffer size n^l_b < N^l then
            continue
        end if
        For each o^l_i in the buffer, compute Â^l_{i^l,t^l} for the t^l-th token of o^l_i (Equation (9))
    end for
    for iteration = 1, ..., μ^l do
        Update the policy model π_θ by maximizing the GRPO+ objective combining with Length Reward K^l
    end for
end for
```

mentations are represented as the following:

In the implementation of Early Preview HGRPO1, the total number of hierarchy is set as 4 for math reasoning problems, in details, $Q^1 < Q^2 < Q^3, Q^4 < Q^3$, $\boldsymbol{Len^1_{max}} < \boldsymbol{Len^2_{max}} < \boldsymbol{Len^3_{max}}, \boldsymbol{Len^2_{max}} \leq \boldsymbol{Len^4_{max}} < \boldsymbol{Len^3_{max}}$. $H^1 \gg H^2 \gg H^3$, $H_3 \sim H_4$.

In the implementation of Early Preview HGRPO2, the total number of hierarchy is set as 4 for math reasoning problems, in details, $Q^1 < Q^2 < Q^3, Q^4 < Q^3$, $\boldsymbol{Len^1_{max}} < \boldsymbol{Len^2_{max}} < \boldsymbol{Len^3_{max}}, \boldsymbol{Len^4_{max}} = \boldsymbol{Len^3_{max}}$. $H^1 \gg H^2 \gg H^3$, $H_3 \sim H_4$.

4 Experiment

To investigate the effectiveness of the proposed two implementations of the preview hierarchical GRPO on the reasoning of LLMs. We conduct a set of experiments in the comparison with the state-of-the art reasoning LLMs models.

4.1 Experiment Setup

We choose DEEPSEEK-R1-DISTILL-QWEN-1.5B (Guo et al., 2025) as our base model, which is a $1.5B$ parameter model and distilled from larger models. We utilize the AdamW (Loshchilov and Hutter, 2019) optimizer with a constant learning rate of 1×10^6 for optimization. For rollout, we set the temperature to 0.6 and sample 16 responses per prompt. In this experiment, we do not utilize a system prompt; instead, we add "Let's think step by step and output the final answer within boxed." at the end of each problem.

4.2 Benchmarks

Math Reasoning Benchmark To better evaluate the trained model, we have selected five benchmarks to assess its performance: MATH 500 (Hendrycks et al., 2021), AIME 2024 (AI-MO, 2024a), AMC 2023 (AI-MO, 2024b), Minerva Math (Lewkowycz et al., 2022), and Olympiad-Bench (He et al., 2024).

4.3 Dataset

Math Reasoning Dataset The training dataset is consisted of 40K problems with three-diffculity level. Particularly, it is consisted of AIME (American Invitational Mathematics Examination) prob-

Table 1: Model Performance Comparison

Model	MATH500	AIME24	AMC	Minerva	OBench	Avg.
Close-Source						
O1-Preview	85.5	44.6	–	–	–	–
O1-Mini	90.0	70.0	–	–	–	–
O1	90.4	71.5	–	–	–	–
Claude 3.7 Sonnet (Standard)	82.2	23.3	–	–	–	–
Open-Source-Large						
DeepSeek-R1	97.3	79.8	–	–	–	–
Qwen3-235B	94.6	85.7	–	–	–	–
Llama 4 Behemoth	95.0	78.0	–	–	–	–
Kimi-1.5	96.2	77.5	–	–	–	–
Qwen 2.5-72B	83.1	30.0	–	–	–	–
Phi4-Reasoning-14B	–	81.3	–	–	–	–
Llama 4 Maverick	18.0	64.0	–	–	–	–
Open-Source-4B/7B						
MIMO-7B	95.8	68.2	–	–	–	–
DeekSeek-7B	92.8	55.5	–	–	–	–
QWEN3-4B	-	73.8	–	–	–	–
Open-Source-1.5B						
DEEPSeek-R1-Distill-QWEN-1.5B	82.8	28.8	62.9	26.5	43.3	48.9
STILL-3-1.5B-Preview	84.4	32.5	66.7	29.0	45.4	51.6
DEEPSCALER-1.5B-Preview	87.8	43.1	73.6	30.2	50.0	57.0
FastCuRL-1.5B-Preview	88.0	43.1	74.2	31.6	50.4	57.5
Ours1-1.5B	88.1	43.2	74.3	31.7	50.4	57.6
Ours2-1.5B	**89.2**	**50.0**	**77.1**	**35.3**	**51.9**	**60.7**

lems (1984-2023), AMC (American Mathematics Competition) problems (prior to 2023), Omni-MATH dataset and Still dataset. For the ranks of particular leaderboard, we split the math reasoning dataset to contain relative sampling according to the particular (Math500,AIME24)leaderboard.

4.4 Evaluation Metric

We set the maximum generation length for the models to 32768 tokens and leverage PASS @1 as the evaluation metric. Specifically, we adopt a sampling temperature of 0.6 and a top-p value of 1.0 to generate k responses for each question, typically $k = 16$.

Specifically, PASS @1 is then calculated as:

$$\text{PASS@1} = \frac{1}{k}\sum_{i=1}^{k} p_i \tag{8}$$

4.5 Math Reasoning Experiments

The proposed hierarchical reasoning model is evaluated against both open-source and closed-source state-of-the-art reasoning models, including O4-Mini, Gemini-2.5-Pro, O3-Mini-2025-01-31, Grok-3-Mini (High), Qwen3-235B-A22B, and others. As shown in Table 3, our 1.5B model achieves impressive performance across multiple benchmarks: 50.0 Pass@1 on AIME24, 89.2 on MATH500, 74.7 on AMC23, 35.3 on Minerva, and 51.9 on OlympiadBench. These results demonstrate the model's robust general reasoning ability across various mathematical and competition-level tasks.

Notably, the hierarchical training strategy enables our 1.5B model to outperform the current best-performing 1.5B reasoning model by 6.9 points on AIME24, 1.4 points on MATH500, 1.1 on AMC23, 4.1 on Minerva, and 1.9 on OlympiadBench—averaging a 3.7-point gain overall. Fur-

Table 2: Combined Model Rankings

MATH-500		AIME	
Model	Accuracy	Model	Accuracy
Gemini 2.5 Pro Exp	95.2%	O3 Mini	86.5%
O3	94.6%	Gemini 2.5 Pro Exp	85.8%
Qwen 3 (235B)	94.6%	O3	85.3%
Grok 3 Mini Fast High Reasoning	94.2%	Grok 3 Mini Fast High Reasoning	85.0%
O4 Mini	94.2%	Qwen 3 (235B)	84.0%
DeepSeek R1	92.2%	O4 Mini	83.7%
O3 Mini	91.8%	DeepSeek R1	74.0%
Gemini 2.5 Flash Preview (Thinking)	91.8%	O1	71.5%
Claude 3.7 Sonnet (Thinking)	91.6%	Grok 3 Mini Fast Low Reasoning	70.6%
Gemini 2.5 Flash Preview	91.6%	Grok 3 Beta	58.7%
O1	90.4%	**Ours-1.5B**	53.3%
Ours-1.5B	90.4%	DeepSeek V3 (03/24/2025)	52.2%
Grok 3 Beta	89.8%	GPT 4.1 mini	49.4%
DeepSeek V3(03/24/2025)	88.6%	Claude 3.7 Sonnet(Thinking)	44.6%
Gemini 2.0 Flash(001)	88.0%	Mistreal Medium 3(05/2025)	42.3%
GPT4.1 Mini	88.0%	GPT4.1	39.8%
GPT4.1	87.2%	Gemini 2.0 Flash(001)	29.8%
Mistreal Medium 3(05/2025)	87.0%	DeepSeek V3	27.5%
LLama4 Maveric	85.2%	GPT4.1 nano	27.3%
Gemini 2.0 Falsh Think Exp	84.6%	LLama 4 Maverick	25.2%
Gemini 1.5 Pro(002)	82.8%	Claude 3.7 Sonnet	22.3%
DeepSeek V3	80.4%	LLama4 Scout	22.3%

thermore, it surpasses several larger parameter models, including O1-Preview, O1-2024-12-17 (Low), O3-Mini-2025-01-31 (Low), and O1-Mini.

On competitive benchmarks, the model ranks 11th on both the Math500 and AIME24 leaderboards, establishing its competitiveness not only among models of similar size but also against larger state-of-the-art LLMs. Particularly, On Math-500, Ours-15B super-passes Grok 3 Beta(89.8%), DeepSeek V3(03/24/2025)(88.6%), Gemini 2.0 Flash(001)(88.0%), GPT4.1 Mini(88.0%), GPT4.1(87.2%), Mistreal Medium 3(05/2025)(87.0%), Gemini 2.0 Falsh Think Exp(84.6%). Similarly, On AIME24, Ours-15B super-passes DeepSeek V3 (03/24/2025)(53.3%), GPT 4.1 mini(49.4 %), Claude 3.7 Sonnet(Thinking)(44.6%), Mistreal Medium 3(05/2025)(42.3%), GPT4.1(39.8%).

5 Discussion

We begin to explore the potential of hierarchical reinforcement learning in enhancing the reasoning capabilities of large language models. By implementing our proposed early preview hierarchical reinforcement learning framework on a relatively limited-scale mathematical dataset, our 1.5B-sized language model demonstrates significant improvements in mathematical reasoning benchmarks. Notably, its performance surpasses the O1-Preview model and approaches the O1-Mini model. Furthermore, on the Math500 and AIME24 mathematical reasoning leaderboards, our model achieves remarkable results, ranking 11th overall. It matches the score of the O1 model on Math500 and secures a position just one rank below Grok 3 Beta on AIME24.

However, we are continuing our exploration of hierarchical reinforcement learning to enhance reasoning capabilities in both small-sized and mid-sized language models. Our focus is on efficiently harnessing small-scale datasets to address math and code reasoning problems. We aim to develop a unified small/mid-sized language model that can achieve competitive scores on both code and math reasoning benchmarks. We plan to release this unified model to the research community, providing a versatile tool for advancing work in mathematical and programming reasoning.

Limitations

This early preview presents an exploratory investigation into hierarchical reinforcement learning, building upon the open-sourced GRPO algorithm. While our initial results are promising, the current version of our work has several important limitations that should be acknowledged to guide future research.

First, our experiments are primarily conducted on datasets focused on mathematical reasoning. This narrow focus restricts the generalizability of our findings to broader domains, such as code reasoning, symbolic logic, and other forms of complex problem-solving. These other areas may involve fundamentally different reasoning dynamics or structural challenges. Consequently, extending our methods to cover a wider range of reasoning tasks across various domains remains an important direction for future work. We believe that rigorous evaluation across diverse task types would help verify the robustness and adaptability of our approach.

Second, due to computational resource constraints, our experiments are conducted on relatively small-scale models with approximately 1.5 billion parameters. While this allows for faster iteration and lower training costs, it potentially limits the scope of our conclusions. Larger models may display qualitatively different learning behaviors, more pronounced performance gains, or even unexpected generalization properties that our current results do not capture. Thus, scaling up the model size and assessing its impact on the effectiveness of hierarchical reinforcement learning methods is a key avenue for future investigation.

Third, our current evaluation framework primarily focuses on task performance metrics in mathematical reasoning scenarios. However, it does not include a detailed analysis of potential societal harms associated with deploying large language models. Issues such as biased output generation, reinforcement of harmful stereotypes, or misuse of models in sensitive applications are critical ethical concerns that remain underexplored in our study. We recognize the significance of these considerations and strongly encourage future work to adopt a more comprehensive and responsible approach that rigorously assesses the social and ethical implications of deploying such models in real-world settings.

Lastly, our evaluation methodology heavily relies on standardized benchmarks that are widely used in the research community. While these benchmarks provide a useful basis for comparison, they may not accurately represent real-world use cases or user preferences, particularly within the context of applied math reasoning tasks. To obtain a more complete understanding of model utility and practical performance, we recommend incorporating human-in-the-loop evaluation protocols and designing domain-specific metrics that better reflect end-user needs and task-specific requirements. Such an approach would facilitate more meaningful insights into the real-world applicability and value of the proposed methods.

References

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

AI-MO. 2024a. Aime 2024. https://huggingface.co/datasets/AI-MO/aimo-validation-aime.

AI-MO. 2024b. Amc 2023. https://huggingface.co/datasets/AI-MO/aimo-validation-amc.

Anthropic. 2025. Claude 3.7 Sonnet and Claude Code. https://www.anthropic.com/claude/sonnet. Accessed 2025-05-13.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*. Abs/2310.12036.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. 2017. The option-critic architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. 2021. Goal-conditioned reinforcement learning with imagined subgoals. In *Advances in Neural Information Processing Systems (NeurIPS)*, page TBD.

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30.

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

Murtaza Dalal, Deepak Pathak, and Ruslan Salakhutdinov. 2021. Accelerating robotic reinforcement learning via parameterized action primitives. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Peter Dayan and Geoffrey E. Hinton. 1993a. Feudal reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 271–278. Morgan Kaufmann Publishers, Inc.

Peter Dayan and Geoffrey E. Hinton. 1993b. Feudal reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 271–278.

Tom Everitt, Marcus Hutter, Ramana Kumar, and Victoria Krakovna. 2021. Reward tampering problems and solutions in reinforcement learning: A causal influence diagram perspective. *arXiv preprint arXiv:2105.00901*.

Tom Everitt, Victoria Krakovna, Laurent Orseau, Marcus Hutter, and Shane Legg. 2017. Reinforcement learning with a corrupted reward channel. *arXiv preprint arXiv:1711.00391*.

Roy Fox, Sanjay Krishnan, Ion Stoica, and Ken Goldberg. 2017. Multi-level discovery of deep options. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1665–1674. PMLR.

Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Aurick Qiao, and Hao Zhang. 2024. Efficiently serving llm reasoning programs with certaindex. *arXiv preprint arXiv:2412.20993*.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, and 1 others. 2024. A framework for few-shot language model evaluation. https://zenodo.org/records/12608602.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. 2025. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*.

X. Guan, L. L. Zhang, Y. Liu, N. Shang, Y. Sun, Y. Zhu, F. Yang, and M. Yang. 2025. Rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*.

Dong Guo, Dongrui Yang, Hao Zhang, Jinjun Song, Rui Zhang, Rui Xu, Qizhe Zhu, Sheng Ma, Peng Wang, Xia Bi, and et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*.

Anna Harutyunyan, Peter Vrancx, Pierre-Luc Bacon, Doina Precup, and Ann Nowé. 2018. Learning with options that terminate off-policy. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pages 3201–3208.

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. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the ACL (Long Papers)*, pages 3828–3850, Bangkok, Thailand.

Dan Hendrycks, Collin Burns, Steven Basart, Antonia Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, Virtual Event. OpenReview.net.

Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Andrew Sendonaris, Gabriel Dulac-Arnold, Ian Osband, John P. Agapiou, Joel Z. Leibo, and Audrunas Gruslys. 2018. Deep q-learning from demonstrations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 3223–3230.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. 2024. V-star: Training verifiers for self-taught reasoners. *arXiv preprint arXiv:2402.06457*.

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

Leslie Pack Kaelbling. 1993. Learning to achieve goals. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1094–1098.

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

Thomas Kipf, Yujia Li, Hanjun Dai, Vinicius Zambaldi, Alvaro Sanchez-Gonzalez, Edward Grefenstette, Pushmeet Kohli, and Peter Battaglia. 2019. Compile: compositional imitation learning and execution. In *International Conference on Machine Learning*, pages 3418–3428. PMLR.

Martin Klissarov, Pierre-Luc Bacon, Jean Harb, and Doina Precup. 2017. Learning options end-to-end for continuous action tasks. In *NeurIPS Hierarchical Reinforcement Learning Workshop*.

Abdul Rahman Kreidieh, Samyak Parajuli, Nathan Lichtlé, Yiling You, Rayyan Nasr, and Alexandre M. Bayen. 2020. Inter-level cooperation in hierarchical reinforcement learning. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pages 2000–2002.

Solomon Kullback and Richard A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Abhinav Kumar, Jaechul Roh, Ali Naseh, Marzena Karpinska, Mohit Iyyer, Amir Houmansadr, and Eugene Bagdasarian. 2025. Overthink: Slowdown attacks on reasoning llms. *arXiv e-prints*, page arXiv:2502.

Andrew Levy, Robert Platt, and Kate Saenko. 2018. Hierarchical actor-critic. In *International Conference on Learning Representations (ICLR)*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems (NeurIPS)*. ArXiv preprint arXiv:2206.14858.

Google LLC. 2025. Gemini 2.5 Flash Preview (Thinking). https://ai.google.dev/gemini-api/docs/changelog. Preview: gemini-2.5-flash-preview-04-17; Accessed: 2025-05-13.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, New Orleans, LA, USA. OpenReview.net.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. S1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

Ofir Nachum, Honglak Lee, Shane Gu, and Sergey Levine. 2018. Data-efficient hierarchical reinforcement learning. *Advances in Neural Information Processing Systems*, 31:3738–3748.

Ashvin Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. 2018. Visual reinforcement learning with imagined goals. In *Advances in Neural Information Processing Systems*, pages 1–13.

Soroush Nasiriany, Huihan Liu, and Yuke Zhu. 2022. Augmenting reinforcement learning with behavior primitives for diverse manipulation tasks. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 7477–7484.

OpenAI. 2024. Learning to reason with llms. https://openai.com/index/learning-to-reason-with-llms/. Accessed 2025-05-13.

OpenAI. 2025a. o3-mini. https://platform.openai.com/docs/models. Accessed: 2025-05-13.

OpenAI. 2025b. o4-mini. https://platform.openai.com/docs/models/o4-mini. Accessed: 2025-05-13.

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

Karl Pertsch, Youngwoon Lee, and Joseph J. Lim. 2020. Accelerating reinforcement learning with learned skill priors. In *Conference on Robot Learning (CoRL)*, pages 944–957.

Qwen Team. 2024. Qwq-32b: Embracing the power of reinforcement learning. https://qwenlm.github.io/blog/qwq-32b-preview/.

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

Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. 2018. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*. Robotics: Science and Systems Foundation.

Sasha Salter, Kristian Hartikainen, Walter Goodwin, and Ingmar Posner. 2022a. Priors, hierarchy, and information asymmetry for skill transfer in reinforcement learning. In *Conference on Robot Learning (CoRL)*, page TBD.

Sasha Salter, Markus Wulfmeier, Dhruva Tirumala, and 1 others. 2022b. Mo2: Model-based offline options. In *Conference on Lifelong Learning Agents*, pages 902–919.

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

Qizhe Shao, Peng Wang, Ronghang Zhu, Rui Xu, Jian Song, Xia Bi, Hao Zhang, Meng Zhang, Yichong Li, and et al. Wu, Yiming. 2024. Deepseek-math: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Konstantinos Shiarlis, Markus Wulfmeier, Shaun Salter, Shimon Whiteson, and Ingmar Posner. 2018. Taco: Learning task decomposition via temporal alignment for control. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 4654–4663. PMLR.

Avi Singh, Huihan Liu, Gaoyue Zhou, Tianhe Yu, Pieter Abbeel, Chelsea Finn, and Sergey Levine. 2021. Parrot: Data-driven behavioral priors for reinforcement learning. In *9th International Conference on Learning Representations (ICLR)*.

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. MIT Press.

Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. 2017. Feudal networks for hierarchical reinforcement learning. In *International Conference on Machine Learning*, pages 3540–3549. PMLR.

P. Wang, L. Li, Z. Shao, R. Xu, D. Dai, Y. Li, D. Chen, Y. Wu, and Z. Sui. 2024. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the ACL (Long Papers)*, pages 9426–9439.

X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou. 2023. Self-consistency improves chain-of-thought reasoning in language models. In *International Conference on Learning Representations*.

J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, and et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Lilian Weng. 2024. Reward hacking in reinforcement learning. `https://lilianweng.github.io/lil-log/2024/11/01/reward-hacking.html`.

XAI. 2024. Grok 3 beta — the age of reasoning agents. `https://x.ai/news/grok-3`.

A. Yang, B. Zhang, B. Hui, B. Gao, B. Yu, C. Li, D. Liu, J. Tu, J. Zhou, J. Lin, and et al. 2024. Qwen 2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Qingcai Yu, Zewei Zhang, Ronghang Zhu, Yilun Yuan, Xiaofu Zuo, Yalong Yue, Tian Fan, Guoliang Liu, Liang Liu, Xiaolin Liu, and et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, Zhenghao Liu, Bowen Zhou, Hao Peng, Zhiyuan Liu, and Maosong Sun. 2024a. Advancing llm reasoning generalists with preference trees. *arXiv preprint*.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. 2024b. Free process rewards without process labels. *arXiv preprint*.

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

Kaiyan Zhang, Sihang Zeng, Ermo Hua, Ning Ding, Zhang-Ren Chen, Zhiyuan Ma, Haoxin Li, Ganqu Cui, Biqing Qi, Xuekai Zhu, Xingtai Lv, Hu Jinfang, Zhiyuan Liu, and Bowen Zhou. 2024a. Ultramedical: Building specialized generalists in biomedicine. *arXiv preprint*.

S. Zhang, Z. Chen, Y. Shen, M. Ding, J. B. Tenenbaum, and C. Gan. 2024b. Planning with large language models for code generation. In *International Conference on Machine Learning*.

Tianren Zhang, Shangqi Guo, Tian Tan, Xiaolin Hu, and Feng Chen. 2020. Generating adjacency-constrained subgoals in hierarchical reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 9814–9826.