Well Begun, Half Done: Reinforcement Learning with Prefix Optimization for LLM Reasoning

Yiliu Sun[1], Zicheng Zhao[1], Yang Wei[1], Yanfang Zhang[1], Chen Gong[2*]

[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China.
[2]School of Automation and Intelligent Sensing, Shanghai Jiao Tong University, Shanghai, China.
Correspondence: chen.gong@sjtu.edu.cn

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) significantly enhances the reasoning capability of Large Language Models (LLMs). Current RLVR approaches typically conduct training across all generated tokens, but neglect to explore which tokens (*e.g.*, prefix tokens) actually contribute to reasoning. This uniform training strategy spends substantial effort on optimizing low-return tokens, which in turn impedes the potential improvement from high-return tokens and reduces overall training effectiveness. To address this issue, we propose a novel RLVR approach called Progressive Prefix-token Policy Optimization (PPPO), which highlights the significance of the prefix segment of generated outputs. Specifically, inspired by the well-established human thinking theory of Path Dependence, where early-stage thoughts substantially constrain subsequent thinking trajectory, we identify an analogous phenomenon in LLM reasoning termed Beginning Lock-in Effect (BLE). PPPO leverages this finding by focusing its optimization objective on the prefix reasoning process of LLMs. This targeted optimization strategy can positively influence subsequent reasoning processes, and ultimately improve final results. To improve the learning effectiveness of LLMs on how to start reasoning with high quality, PPPO introduces two training strategies: (a) Progressive Prefix Retention, which shapes a progressive learning process by increasing the proportion of retained prefix tokens during training; (b) Continuation Accumulated Reward, which mitigates reward bias by sampling multiple continuations for one prefix token sequence, and accumulating their scores as the reward signal. Extensive experimental results on various reasoning tasks (*e.g.*, math, physics, chemistry, biology) demonstrate that our proposed PPPO outperforms representative RLVR methods, with the accuracy improvements of 18.02% on only 26.17% training tokens.

Introduction

Since the advance of Large Language Models (LLMs), reasoning has emerged as one of their core capabilities (Lyu et al. 2025; Wenjuan et al. 2025; Talmor et al. 2019; Yu et al. 2020; Qin et al. 2025; Sun et al. 2025b; Zhao et al. 2023; Zhang et al. 2025b; Zhao et al. 2024; Wang et al. 2025a; Sun et al. 2025a; Ding et al. 2025), which demonstrates a promising path towards Artificial General Intelli-

gence (AGI) (OpenAI et al. 2025). Therefore, it is critical to analyze and improve the reasoning capability of LLMs.

Recent advancements in LLMs, such as DeepSeek R1 (Guo et al. 2025) and OpenAI o1 (OpenAI 2024), have demonstrated promising progress on various reasoning benchmarks. One of the key factors to drive these improvements is Reinforcement Learning with Verifiable Rewards (RLVR) (Lambert et al. 2025; Shao et al. 2024; Xie et al. 2025; Zheng et al. 2025; Zhang et al. 2025a; Lou et al. 2025). Generally, RLVR adopts rule-based verification functions (*e.g.*, correctness check (Guo et al. 2025; Yu et al. 2025), semantic entropy (Zhao et al. 2025)) to provide reward signals for sampled outputs, and then optimizes model outputs towards high quality by encouraging generations of the content with high reward signals.

However, current RLVR approaches face the challenge of limited training effectiveness. Specifically, most existing methods directly train across all generated tokens indiscriminately, but neglect the variable contributions of different tokens to the final performance. Given that tokens play heterogeneous roles in reasoning processes, this uniform treatment spends considerable effort on optimizing low-impact tokens, which in turn reduces training effectiveness and hinders further performance improvement brought by critical tokens (Wang et al. 2025c; Gu et al. 2024).

To address this issue, this paper proposes **P**rogressive **P**refix-token **P**olicy **O**ptimization (PPPO), a novel RLVR method inspired by the well-established human thinking theory of **Path Dependence** (David 1975). As revealed in (David 1975), during the human thinking process, the initial thought pattern substantially constrains the subsequent thinking trajectory, which in turn significantly influences the final result. This theory highlights the foundational influence of initial thoughts on the entire thinking process.

Given the autoregressive generation nature of LLMs, where each token is generated based on previous ones, we explore whether path dependence manifests in LLM reasoning (see the Preliminary section for details). Through extensive empirical investigations, we identify an analogous phenomenon termed **B**eginning **L**ock-in **E**ffect (BLE), where the initial reasoning process significantly constrains the subsequent reasoning steps and influences final results. For instance, as illustrated in Figure 1a, the flawed strategy adopted by the initial reasoning (*e.g.*, decimal truncation or
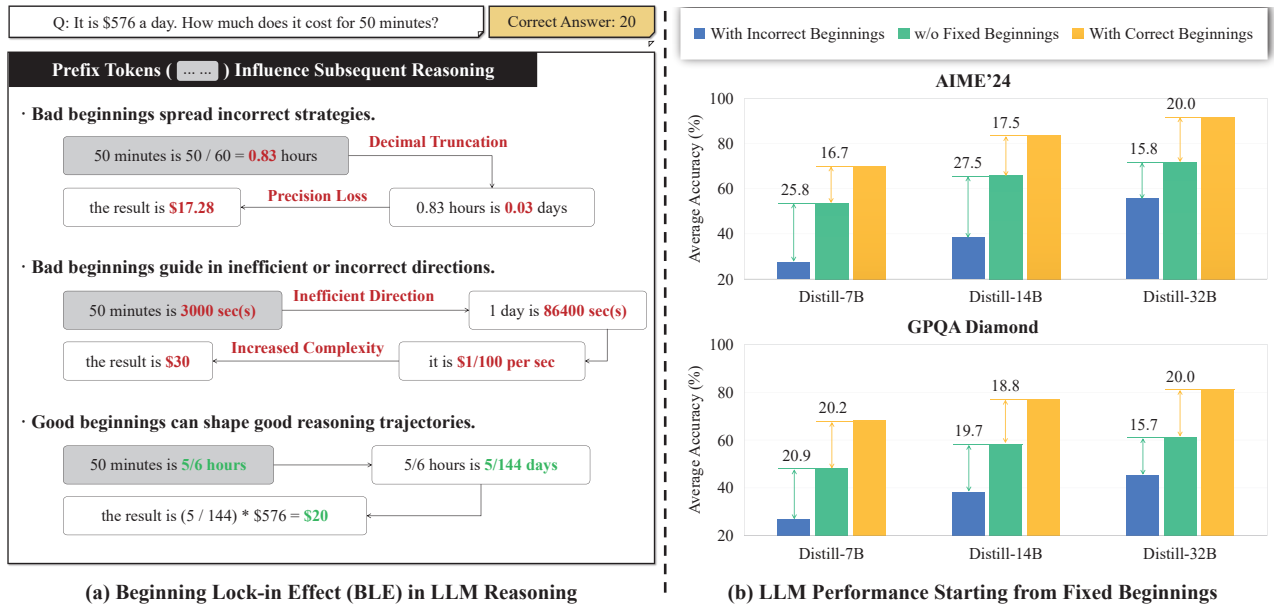
Figure 1: (a) In LLM reasoning, we identify a critical phenomenon termed BLE, where the initial reasoning significantly affects the subsequent reasoning. (b) Reasoning from prefix tokens extracted from correct outputs significantly improves the reasoning performance, while prefix tokens extracted from incorrect outputs lead to substantial performance degradation.

unnecessary unit conversion) misleads the reasoning process into an incorrect (precision loss) or inefficient (increased complexity) trajectory, and results in an incorrect result.

Based on this insight, our proposed PPPO improves the effectiveness of RLVR training by applying policy updates exclusively to prefix tokens (*i.e.*, the first small part of an output token sequence) while masking the gradients of subsequent tokens. During the PPPO training, two key training strategies are incorporated, including **Progressive Prefix Retention** and **Continuation Accumulated Reward**. The former one forms a progressive learning process from simple to complex by gradually increasing the fraction of retained prefix tokens during PPPO training. The latter one requires the trained LLM to generate multiple continuations from a fixed prefix token sequence extracted from a full output, and then accumulates their scores as the reward signal for that prefix token sequence. By explicitly teaching LLMs to start reasoning with high quality, PPPO steers the entire reasoning trajectory toward better reasoning performance.

The effectiveness of PPPO has been well confirmed by extensive experiments on five complex reasoning benchmarks. For instance, when compared with the representative RLVR methods, PPPO increases the task accuracy by up to 18.02% on a single benchmark (AIME'25), and increases the average accuracy across all the benchmarks by up to 14.64%. Additionally, PPPO results in efficient reasoning, which reduces output tokens by up to 18.35% while maintaining high accuracy. These experimental results demonstrate that our PPPO enables LLMs to start reasoning with high quality, which can lead to correct and concise reasoning trajectories.

The main contributions are summarized as follows:

- We identify a critical phenomenon in LLM reasoning

termed **Beginning Lock-in Effect (BLE)**, which demonstrates that the prefix tokens significantly shape subsequent reasoning trajectories.

- We propose a novel RLVR method termed **Progressive Prefix-token Policy Optimization (PPPO)**. Building upon BLE, PPPO can improve the training effectiveness of RLVR and enhance the reasoning capability of LLMs by focusing on the optimization of prefix tokens.

- We introduce two key training strategies to improve the learning effectiveness of LLMs during PPPO training, including **Progressive Prefix Retention** and **Continuation Accumulated Reward**.

- We conduct extensive experiments on various reasoning benchmarks to show that **our PPPO consistently outperforms representative RLVR baseline methods** (*i.e.*, GRPO, DAPO, INTUITOR, and DAPO-FT).

Related Work

Reinforcement learning with verifiable rewards (Qu et al. 2025; Rafailov et al. 2023; Byun et al. 2024; He et al. 2024; Zhang et al. 2024) has been a promising framework to enhance the reasoning capability of LLMs. Typically, RLVR fine-tunes LLMs by using reward signals derived from human preferences or automatic task-specific validations to improve the quality of LLM outputs. For example, Schulman et al. (2017) introduce Proximal Policy Optimization (PPO), which adopts Kullback-Leibler Divergence to constrain the magnitude of parameter updates and improve the training stability. Notably, PPO needs to train a critic model to estimate the advantages of LLM outputs, which is costly. Hence, Shao et al. (2024) propose Group Relative Policy Optimization (GRPO), which foregoes the critic model and

estimates the advantage of each output according to the average score of all the outputs instead. Encouraged by the success of GRPO, many RLVR methods that do not require these costly models have been proposed (Yeo et al. 2025; Xie et al. 2025; Zheng et al. 2025; Xu et al. 2025; Yue et al. 2025; Zhang et al. 2025a). For instance, Yu et al. (2025) introduce Dynamic Sampling Policy Optimization (DAPO) to alleviate four issues (*i.e.*, entropy collapse, training instability, low training efficiency, and reward hacking) faced by GRPO. Zheng et al. (2025) propose GRPO with Efficient Selective Rollout (GRESO), which improves the quality of sampled problems and thus enhances the training efficiency.

However, existing RLVR methods suffer from low training effectiveness (Sheng et al. 2025; Wang et al. 2025c; Zheng et al. 2025). Specifically, these methods train across all tokens, but overlook the heterogeneous contributions of different tokens to reasoning performance. As a result, much effort is spent on optimizing tokens with low or negative returns. Although Wang et al. (2025c) attempt to address this issue by solely optimizing high-entropy tokens (*e.g.*, wait, however, rethink), this strategy encourages LLMs to explore, but cannot guarantee the quality of the reasoning paths triggered by these high-entropy tokens. As a result, the improvement brought by this strategy is unstable.

Distinctively, inspired by the human thinking theory of Path Dependence, our PPPO improves the training effectiveness of RLVR by focusing on optimizing prefix tokens, which play critical roles in shaping the reasoning trajectory. Hence, by learning to start reasoning effectively, LLMs can maintain a coherent and high-quality reasoning trajectory, which in turn improves the final performance.

Preliminaries

In this section, we first introduce the concept of Path Dependence in human thinking processes and then explore its counterpart in the reasoning processes of LLMs.

Path Dependence

Path dependence (David 1975) is a well-established concept of human thinking patterns, where early-stage thoughts constrain subsequent thoughts. Based on the effects on subsequent thoughts, it can be categorized as either benign or malignant. A benign one means that high-quality initial thoughts steer the thinking process toward a desirable outcome, while a malignant one indicates that low-quality initial thoughts result in a suboptimal outcome. Namely, initial thoughts significantly influence long-term outcomes, which suggests that high-quality initial thoughts increase the likelihood of success (Page et al. 2006).
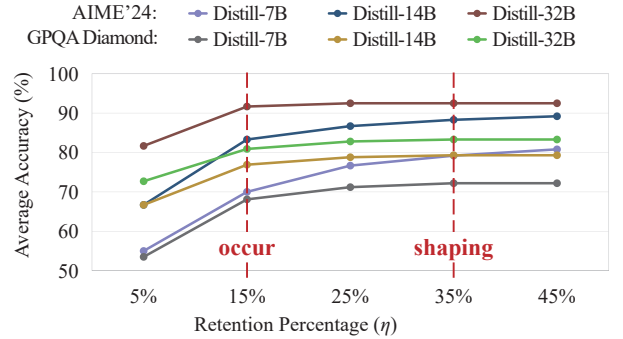
Beginning Lock-in Effect

Due to the autoregressive nature of LLMs, where each token is generated based on previous tokens, a critical research question is raised: *Does the quality of prefix tokens exert a significant influence on the final result in LLM reasoning?*

To answer this question, we design controlled experiments, in which LLMs are required to reason from fixed prefix tokens. Specifically, for each problem in two reasoning
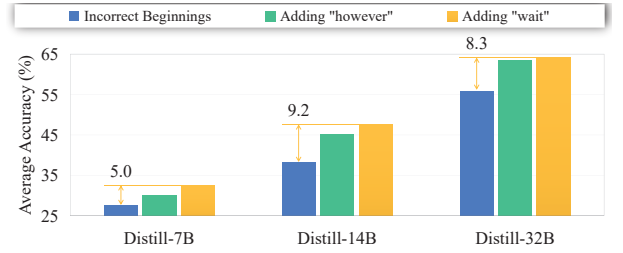
(a) LLMs struggle to escape from the trajectories shaped by incorrect prefix token sequences.

(b) As the length of the prefix token sequence increases, BLE gradually establishes.

Figure 2: Explorations of beginning lock-in effect.

benchmarks (*i.e.*, AIME'24 (MAA 2024) and GPQA Diamond (Rein et al. 2023)), we employ DeepSeek-R1-Distill-Qwen series (Guo et al. 2025) to generate 4 correct and incorrect outputs, respectively. We then extract the first 15% of the tokens from each output as a fixed prefix token sequence. For each fixed prefix token sequence, we prompt the original LLM to generate eight continuations and calculate the average accuracy. As reported in Figure 1b, we observe that all the evaluated LLMs exhibit a significant performance decline (up to 27.5% on AIME'24 with DeepSeek-R1-Distill-Qwen-14B) when reasoning from the fixed prefix token sequences sampled from incorrect answers, while demonstrating substantial improvement (up to 20.2%) when initialized with the fixed prefix token sequences from correct answers.

To test whether LLMs can recover from the low-quality initial reasoning, we add "wait" and "however" after those malignant prefix tokens, which can encourage LLMs to reflect previous reasoning (Muennighoff et al. 2025; Wang et al. 2025b). As shown in Figure 2a, even with these corrective interventions, the maximum recovery of accuracy reaches only 9.2%. The results reveal that LLMs struggle to escape from the flawed initial reasoning, and further support the path dependence hypothesis in LLM reasoning. We term this phenomenon as **B**eginning **L**ock-in **E**ffect (BLE).

Methodology

This section introduces the overall framework of our proposed **P**rogressive **P**refix-token **P**olicy **O**ptimization (PPPO). As described in the Preliminaries section, high-
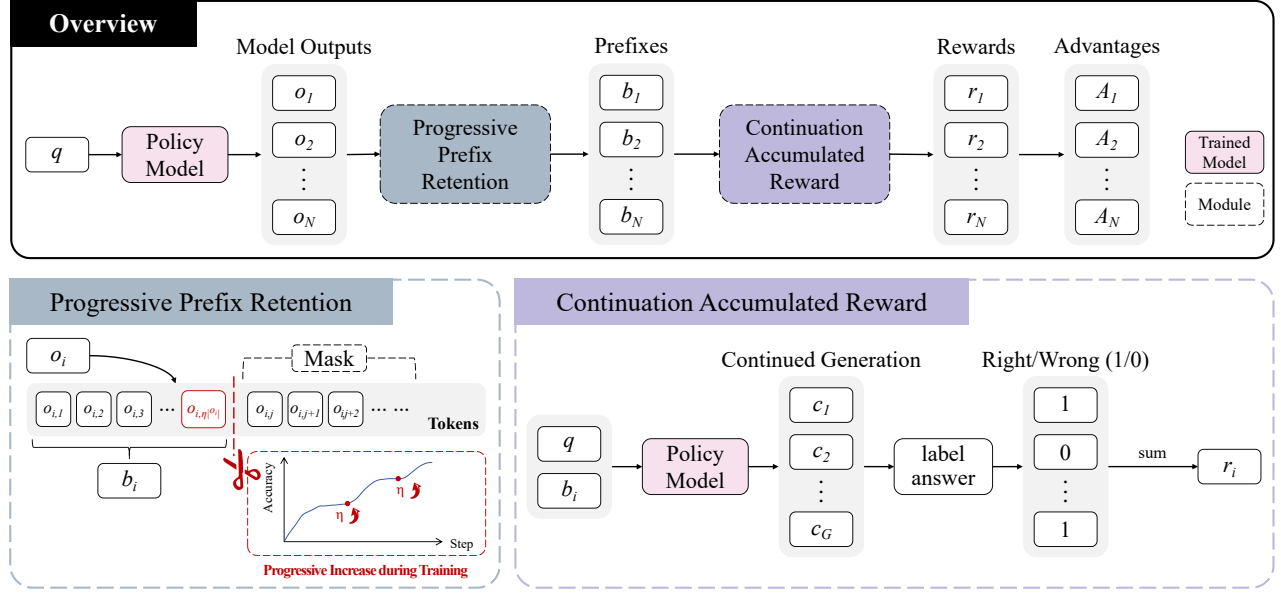
Figure 3: Illustration of PPPO, which is inspired by the human thinking theory of Path Dependence. PPPO focuses on optimizing the prefix segments of LLM outputs and thus positively influences the subsequent reasoning paths. During the PPPO training, two strategies are introduced, including "Progressive Prefix Retention" and "Continuation Accumulated Reward".

quality initial reasoning paths yield high-quality results, while low-quality initial reasoning paths harm model performance. This phenomenon (BLE) can be expressed by the probability formula as follows:

$$P\left(\hat{y}=a \mid q, I_{high-quality}\right) > P\left(\hat{y}=a \mid q\right) \quad (1)$$
$$> P\left(\hat{y}=a \mid q, I_{low-quality}\right),$$

where $\hat{y}$, q, a, and I denote an answer of the LLM, a question, the correct answer, and an initial reasoning path, respectively. Inspired by BLE, PPPO enhances LLM reasoning by explicitly optimizing the initial stages of reasoning processes. As illustrated in Figure 3, PPPO retains the gradients of the first small portion of generated tokens, and masks those of subsequent tokens. This strategy enables LLMs to focus on learning high-quality reasoning initiations, which in turn improves the subsequent reasoning performance.

Specifically, for q and its correct answer a in the dataset $\mathcal{D}$, PPPO samples N outputs $\{\mathbf{o}_i\}_{i=1}^N$ from the policy model π_θ, and retains their first η of tokens to obtain N prefix sequences $\{\mathbf{b}_i\}_{i=1}^N$, where η is a percentage. Starting from each $\mathbf{b}_i$, PPPO samples G outputs $\{\mathbf{c}_j\}_{j=1}^G$. Subsequently, PPPO optimizes the policy via the following objective:

$$\mathcal{J}_{PPPO}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\{\mathbf{o}_i\}_{i=1}^N\sim\pi_{\theta_{old}}(\cdot|q),\{\mathbf{c}_j\}_{j=1}^G\sim\pi_{\theta_{old}}(\cdot|q,\mathbf{b}_i)}$$

$$\left[\frac{1}{\sum_{k=1}^N |\mathbf{o}_k|} \sum_{i=1}^N \sum_{j=1}^{|\mathbf{o}_i|} H(j, \mathbf{o}_i) \cdot \min\left(r_{i,j}(\theta)\hat{A}_{i,j}, \right.\right.$$

$$\left.\left. \text{clip}\left(r_{i,j}(\theta), 1-\varepsilon_{low}, 1+\varepsilon_{high}\right)\hat{A}_{i,j}\right)\right], \quad (2)$$

where

$$r_{i,j}(\theta) = \frac{\pi_\theta\left(o_{i,j} \mid q, \mathbf{o}_{i,<j}\right)}{\pi_{\theta_{old}}\left(o_{i,j} \mid q, \mathbf{o}_{i,<j}\right)}, \quad (3)$$

$$\hat{A}_{i,j} = \frac{R_i - \text{mean}\left(\{R_i\}_{i=1}^N\right)}{\text{std}\left(\{R_i\}_{i=1}^N\right)}. \quad (4)$$

In above equations, $H(\cdot, \cdot)$ retrains the advantages of prefix tokens which will be detailed in Eq. (5), clip $(\cdot, \cdot, \cdot)$ denotes a constraint function for policy update, mean $(\cdot)$ denotes a function which calculates the average value, std $(\cdot)$ denotes a function which calculates the standard deviation, $|\mathbf{o}_i|$ returns the token number of $\mathbf{o}_i$, ε is the clipping range of the importance sampling ratio, and R_i is the reward of $\mathbf{o}_i$. The algorithm can be found in Algorithm 1. To enhance the learning effectiveness on starting reasoning with high quality, PPPO incorporates two key strategies, including Progressive Prefix Retention and Continuation Accumulated Reward, which are described in the following subsections.

Progressive Prefix Retention

For each $\mathbf{o}_i$, PPPO retains the gradients of the first η tokens for optimization and masks those of the remaining tokens, which can be achieved by $H(\cdot, \cdot)$:

$$H(j, \mathbf{o}_i) = \mathbb{1}\left(j \le \lfloor \eta \cdot |\mathbf{o}_i| \rfloor\right), \quad (5)$$

where $\mathbb{1}(\cdot)$ and $\lfloor \cdot \rfloor$ denote an indicator function and a function which rounds a value down to the nearest integer, respectively. During the training, PPPO gradually increases η.

Algorithm 1: Training Process of PPPO

Input: Initial policy model π_θ; reward model R; task set $\mathcal{D}$; hyperparameters ε_{low}, ε_{high}; initial sequence proportion η; maximum training steps S

Output: Optimized policy model π_θ

1 **for** $step = 1, 2, \ldots, S$ **do**
2   Sample a batch of tasks $\mathcal{D}_d \sim \mathcal{D}$;
3   Set old policy parameters $\pi_{\theta_{old}} \leftarrow \pi_\theta$;
4   **for** each query $q \in \mathcal{D}_d$ **do**
5   Sample N outputs $\{\mathbf{o}_i\}_{i=1}^N$ from policy π_θ;
6   Retain the first η tokens of each output to form prefix token sequences $\{\mathbf{b}_i\}_{i=1}^N$;
7   **for** each sequence $\mathbf{b}_i \in \{\mathbf{b}_i\}_{i=1}^N$ **do**
8   Sample $\{\mathbf{c}_j\}_{j=1}^G$ starting from q and $\mathbf{b}_i$;
9   **end**
10   Compute rewards $\{\mathbf{r}_i\}_{i=1}^N$ for $\{\mathbf{b}_i\}_{i=1}^N$ using reward model R;
11   **end**
12   **for** each sequence $\mathbf{b}_i$ **do**
13   Compute the advantage $\hat{A}_{i,j}$ for each token $b_{i,j}$ within sequence $\mathbf{b}_i$;
14   **end**
15   Update θ by using calculated advantages $\hat{\mathbf{A}}$;
16   **if** performance improvement of π_θ reaches a bottleneck **then**
17   Increase η for subsequent iterations;
18   **end**
19 **end**
20 **return** Optimized policy model π_θ;

The increasing trigger occurs when the LLM performance stagnates, which is defined as the absence of the accuracy increase on the validation set after multiple training steps:

$$\eta = \begin{cases} \eta, & \text{if } \Delta acc > 0 \\ \eta + \Delta\eta, & \text{if } \Delta acc \leq 0 \end{cases}, \quad (6)$$

where Δacc and $\Delta\eta$ denote the accuracy change and the increment step of η, respectively. This strategy forms a progressive learning process from simple to complex. Specifically, PPPO starts with short prefix token sequences, which reduces the learning complexity and allows LLMs to rapidly develop core competencies of starting reasoning with high quality. Subsequently, PPPO gradually increases the optimized sequence length. Since the model has built a solid foundation before, it can maintain high learning quality and stability after increasing the optimized sequence length.

Continuation Accumulated Reward

Most existing RLVR methods use single-sample reward signals for output optimization, such as correctness verifications and format checks:

$$R_i^{correct} = \mathbb{1}\left(\hat{y}_{\mathbf{o}_i} = a\right), \quad R_i^{format} = \mathbb{1}\left(str \subseteq \mathbf{o}_i\right), \quad (7)$$

where $\hat{y}_{\mathbf{o}_i}$ denotes the answer extracted from $\mathbf{o}_i$ and *str* donates a string representing a specific reasoning structure. However, for prefix token sequences, the single-sample evaluation suffers from high stochasticity, which leads to biased reward signals and unstable policy updates.

To alleviate this problem, PPPO introduces Continuation Accumulated Reward to evaluate prefix tokens. Specifically, for each $\mathbf{o}_i = (o_{i,1}, o_{i,2}, \ldots, o_{i,|\mathbf{o}_i|})$, PPPO retains the first η of tokens to form $\mathbf{b}_i$:

$$\mathbf{b}_i = \left\{o_{i,j}\right\}_{j=1}^{\lfloor \eta \cdot |\mathbf{o}_i| \rfloor}. \quad (8)$$

For each $\mathbf{b}_i$, PPPO samples G continuations $\{\mathbf{c}_j\}_{j=1}^G$ conditioned on $\mathbf{b}_i$ and the corresponding question q:

$$P\left(\mathbf{c}_j \mid q, \mathbf{b}_i\right) = \prod_k P\left(c_{j,k} \mid \mathbf{c}_{j,<k}, q, \mathbf{b}_i\right). \quad (9)$$

The reward R_i for $\mathbf{b}_i$ is computed as the cumulative score of the continuations $\{\mathbf{c}_j\}_{j=1}^G$ and the original output $\mathbf{o}_i$:

$$R_i = \sum_{j=1}^G \mathbb{1}\left(\hat{y}_{\mathbf{c}_j} = a\right) + \mathbb{1}\left(\hat{y}_{\mathbf{o}_i} = a\right). \quad (10)$$

This accumulated reward mechanism mitigates randomness and provides reliable evaluations for prefix tokens.

Experiments

In this section, we introduce the experimental setup and analyses of our experimental results.

Experimental Setup

In this part, we provide the experimental setup, including training details, baseline methods, and evaluation strategies.

Training Details. We adopt the Qwen3 series (*i.e.*, Qwen3-1.7B, Qwen3-4B and Qwen3-8B) (Yang et al. 2025) under the thinking mode as backbone models, and DAPO-Math-17K as the training dataset. For each question q, we sample 8 outputs $\{\mathbf{o}_i\}_{i=1}^8$. For each prefix token sequence $\mathbf{b}_i$, we sample 8 continuations $\{\mathbf{c}_j\}_{j=1}^8$. Additionally, the models are trained with a learning rate of 1×10^{-6}. By following (Yu et al. 2025), we set ε_{low} to 0.2 and ε_{high} to 0.28.

Moreover, as shown in Figure 2b, model performance exhibits a significant transition when the retention proportion η reaches 15%, and stabilizes when η reaches 35%. This case indicates that BLE manifests within the first 15% of tokens, and essentially establishes by 35%. Accordingly, the retention proportion η in PPPO is initialized at 15% and gradually increased to 35% by steps of 5%.

Baseline Methods. We compare our PPPO with 5 methods: (a) Backbone Models, (b) GRPO (Shao et al. 2024), (c) DAPO (Yu et al. 2025), (d) INTUITOR (Zhao et al. 2025), and (e) DAPO with Forking Tokens (DAPO-FT) (Wang et al. 2025c). For fair comparison, the maximum response length of LLMs is 10240 tokens in all the methods. Moreover, for each training step, PPPO samples 8×8 outputs and all the baseline methods sample 64 outputs.

Method	AIME'24	AIME'25	MATH 500	AMC'23	GPQA Diamond	Average
Qwen3-1.7B	25.83	18.33	76.05	56.93	28.16	41.06
+ GRPO (Guo et al. 2025)	27.50	20.00	82.10	60.24	29.80	43.93
+ DAPO (Yu et al. 2025)	30.83	23.33	83.60	**63.03**	<u>31.82</u>	46.52
+ INTUITOR (Zhao et al. 2025)	27.29	17.71	78.13	60.99	29.96	42.82
+ DAPO-FT (Wang et al. 2025c)	<u>31.25</u>	<u>23.96</u>	<u>83.93</u>	62.80	31.44	<u>46.68</u>
+ PPPO (**ours**)	**38.65**	**28.96**	**87.80**	<u>62.89</u>	**39.52**	**51.58**
Qwen3-4B	48.75	35.42	84.46	72.67	43.59	56.98
+ GRPO (Guo et al. 2025)	52.08	37.71	88.40	76.77	46.78	60.35
+ DAPO (Yu et al. 2025)	<u>56.46</u>	<u>42.08</u>	92.33	81.63	<u>49.37</u>	64.37
+ INTUITOR (Zhao et al. 2025)	51.04	35.42	88.26	75.83	46.43	59.40
+ DAPO-FT (Wang et al. 2025c)	56.25	<u>42.08</u>	<u>92.38</u>	<u>82.00</u>	49.21	<u>64.38</u>
+ PPPO (**ours**)	**63.54**	**53.44**	**94.60**	**83.06**	**52.07**	**69.34**
Qwen3-8B	52.29	38.75	86.06	75.08	46.12	59.66
+ GRPO (Guo et al. 2025)	58.75	42.29	91.00	79.44	50.51	64.40
+ DAPO (Yu et al. 2025)	63.13	48.75	93.21	83.96	<u>55.18</u>	68.85
+ INTUITOR (Zhao et al. 2025)	55.42	40.83	91.20	78.24	49.31	63.00
+ DAPO-FT (Wang et al. 2025c)	<u>63.75</u>	<u>49.38</u>	<u>93.65</u>	<u>84.11</u>	54.77	<u>69.13</u>
+ PPPO (**ours**)	**72.19**	**59.69**	**94.73**	**86.75**	**58.13**	**74.30**

Table 1: Comparison results on 5 benchmarks. The unit of all the results is "%". The best and second-best scores are shown in bold and underlined, respectively.

Evaluation Strategies. We evaluate our method and the baseline methods on 5 benchmarks, which are widely used to evaluate the reasoning capability of LLMs: (a) AIME'24 (MAA 2024), (b) AIME'25 (Balunović et al. 2025), (c) MATH 500 (Hendrycks et al. 2021), (d) AMC'23 (MAA 2023), and (e) GPQA Diamond (Rein et al. 2023). By following (Yue et al. 2025; Wang et al. 2025c), for each dataset, we test 32 times under a zero-shot setting and report the average accuracy, *avg@32*.

Main Results

In this part, we present the experimental results and detailed analyses to highlight the effectiveness of our PPPO.

PPPO outperforms baseline methods. As shown in Table 1, PPPO achieves the highest accuracy in 14 out of 15 settings (3 backbone models × 5 benchmarks). The only exception is on the AMC'23 benchmark with the Qwen3-1.7B model, where PPPO slightly underperforms DAPO by 0.14%. In all other cases, PPPO outperforms all the baseline methods (increasing accuracy by up to 18.02%). These results demonstrate the consistent superiority of our proposed PPPO in enhancing the reasoning capability of LLMs. Additionally, we define the metric of LE as the ratio of Average Accuracy Increase (AAI) to Proportion of Optimized Tokens (POT) during RLVR training. As reported in Table 2, PPPO achieves the highest LE compared with all the baseline methods, which delivers accuracy improvements of up to 14.64% while optimizing only 24.83% of the generated tokens. The results reveal that PPPO enables LLMs to learn key reasoning abilities (*e.g.*, high-quality reasoning initiation, effective reasoning strategies) from a few tokens, which further confirm the superior training effectiveness of PPPO.

PPPO results in efficient reasoning. For each backbone

Qwen3	Method	AAI ↑	POT ↓	LE ↑
1.7B	GRPO	2.87	100.00	2.87
	DAPO	5.46	100.00	5.46
	INTUITOR	1.76	100.00	1.76
	DAPO-FT	<u>5.62</u>	**20.00**	<u>28.08</u>
	PPPO (**ours**)	**10.52**	<u>29.19</u>	**36.05**
4B	GRPO	3.37	100.00	3.37
	DAPO	<u>7.39</u>	100.00	7.39
	INTUITOR	2.42	100.00	2.42
	DAPO-FT	<u>7.39</u>	**20.00**	<u>37.02</u>
	PPPO (**ours**)	**12.36**	<u>26.17</u>	**47.24**
8B	GRPO	4.74	100.00	4.74
	DAPO	9.19	100.00	9.19
	INTUITOR	3.34	100.00	3.34
	DAPO-FT	<u>9.47</u>	**20.00**	<u>47.36</u>
	PPPO (**ours**)	**14.64**	<u>24.83</u>	**58.95**

Table 2: Comparison of training effectiveness. Here, "↑" means that higher values are better, and "↓" means that lower values are better. The best and second-best scores are shown in bold and underlined, respectively.

model, we calculate the average token number of its outputs after being trained on each baseline method, and present the results in Figure 4. While achieving the highest accuracies, PPPO achieves the lowest reasoning cost compared with other baseline methods, which reduces the average number of generated tokens by up to 18.35%. The reason is that PPPO enables LLMs to start reasoning with high quality, which leads to concise yet effective reasoning trajectories.
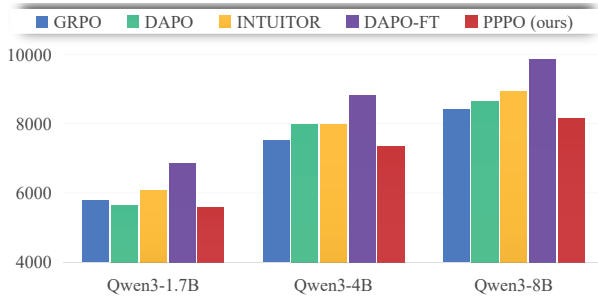
Figure 4: Average output lengths of the backbone models after being trained on each baseline method.
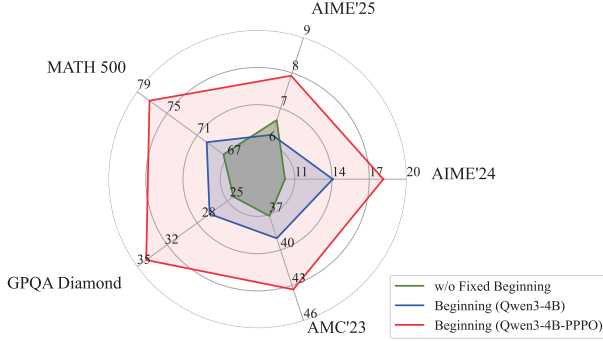

Figure 5: Task performance of Qwen2.5-7B-Instruct when reasoning from prefix tokens extracted from Qwen3-4B with and without PPPO training.

Performance Investigation

In this part, we conduct in-depth investigations on our proposed PPPO to analyze its effectiveness.

PPPO enables high-quality reasoning initiation. We require Qwen2.5-7B-Instruct (Qwen et al. 2025) to reason starting from prefix tokens extracted from Qwen3-4B with and without PPPO training. As shown in Figure 5, the prefix tokens extracted from untrained Qwen3-4B provide only marginal performance gains for Qwen2.5-7B-Instruct, and even reduce accuracy on AIME'25. In contrast, starting reasoning from the prefix tokens extracted from Qwen3-4B with PPPO training enables Qwen2.5-7B-Instruct to achieve optimal performance across all benchmarks, which improves accuracy by up to 9.04%. The results confirm that our proposed PPPO enables Qwen3-4B to start reasoning with high quality. As a result, the prefix tokens generated by Qwen3-4B with PPPO training can guide the reasoning process towards a desirable trajectory.

Continuation accumulated reward enhances training effectiveness and stability. By using Qwen3-4B as the backbone model, we implement PPPO with various generation times G. For each G, we train 4 times and report the average accuracies (*avg@4*) and accuracy variances (*var@4*) in Table 3. We can find that the single-sample reward strategy yields low accuracy and high variance. In contrast, the reward mechanism adopted by PPPO increases the average accuracy by up to 8.37% and reduces the accuracy variance

Method	Sample Strategy	G	avg@4 ↑	var@4 ↓
PPPO	Single	1	60.46	3.30
	Multiple	4	66.11	1.47
		8	69.36	0.63
		16	69.53	0.56

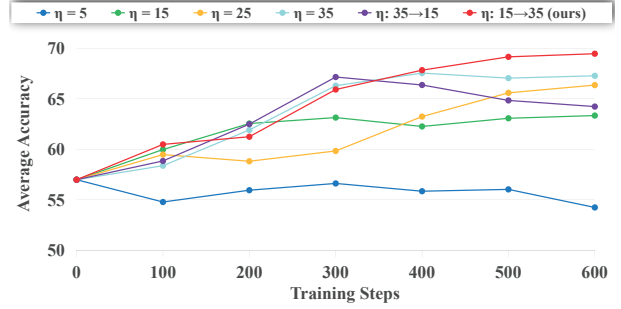Table 3: Results under different sampling times G.

Figure 6: Training performance of PPPO with different prefix retention strategies. The backbone model is Qwen3-4B.

by up to 2.67%. The reason is that our reward mechanism mitigates the randomness of single sampling through multiple explorations, which enables accurate evaluation for prefix tokens.

Progressive prefix retention strategy improves learning quality. To demonstrate the superiority of our progressive prefix retention strategy, we compare it with 5 different prefix retention strategies. As shown in Figure 6, our progressive prefix retention strategy stabilizes the learning process of the model and increases the average accuracy by up to 12.49%. Furthermore, compared with the strategy of "$\eta = 35$", our strategy achieves better learning effectiveness (increasing the average accuracy by 2.19% and reducing the number of training tokens by 8.83%).

Conclusion

In this work, we introduce a critical yet underexplored phenomenon, Beginning Lock-in Effect, in LLM reasoning. It reveals that the quality of initial reasoning steps exerts a significant influence on the reasoning trajectory and outcome. By leveraging this insight, we propose a novel RLVR method termed Progressive Prefix-token Policy Optimization (PPPO). PPPO shifts the focus of RLVR training from uniform all-token optimization to a trajectory-specific optimization, where prefix tokens are optimized independently due to their disproportionate impact on reasoning processes. Comprehensive experiments across diverse reasoning tasks confirm that PPPO shows superior performance to various representative RLVR methods. Beyond methodological advancements, our findings highlight the importance of trajectory-specific optimization and indicate promising avenues for integrating cognitive insights into LLM research, which ultimately contributes to the development of more robust and human-aligned LLM systems.

Acknowledgements

References

Balunović, M.; Dekoninck, J.; Petrov, I.; Jovanović, N.; and Vechev, M. 2025. MathArena: Evaluating LLMs on Uncontaminated Math Competitions. arXiv:2505.23281.

Byun, J.-S.; Chun, J.; Kil, J.; and Perrault, A. 2024. ARES: Alternating Reinforcement Learning and Supervised Fine-Tuning for Enhanced Multi-Modal Chain-of-Thought Reasoning Through Diverse AI Feedback. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*.

David, P. A. 1975. *Technical choice innovation and economic growth: essays on American and British experience in the nineteenth century*. Cambridge University Press.

Ding, Y.; Lyu, H.; Cao, Y.; Song, W.; Tian, L.; and Liu, S. 2025. Similar Cluster Frequency Entropy: A Novel Uncertainty Estimator for Detecting Large Language Models Confabulations. *Chinese Journal of Electronics*.

Gu, S.; Shi, L.; Ding, Y.; Knoll, A.; Spanos, C. J.; Wierman, A.; and Jin, M. 2024. Enhancing efficiency of safe reinforcement learning via sample manipulation. *Advances in Neural Information Processing Systems*, 37: 17247–17285.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv:2501.12948.

He, T.; Liao, L.; Cao, Y.; Liu, Y.; Liu, M.; Chen, Z.; and Qin, B. 2024. Planning Like Human: A Dual-process Framework for Dialogue Planning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In *35th Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Lambert, N.; Morrison, J.; Pyatkin, V.; Huang, S.; Ivison, H.; Brahman, F.; Miranda, L. J. V.; Liu, A.; Dziri, N.; Lyu, S.; Gu, Y.; Malik, S.; Graf, V.; Hwang, J. D.; Yang, J.; Bras, R. L.; Tafjord, O.; Wilhelm, C.; Soldaini, L.; Smith, N. A.; Wang, Y.; Dasigi, P.; and Hajishirzi, H. 2025. Tulu 3: Pushing Frontiers in Open Language Model Post-Training. arXiv:2411.15124.

Lou, X.; Zhang, J.; Xie, J.; Liu, L.; Yan, D.; and Huang, K. 2025. Sequential Preference Optimization: Multi-Dimensional Preference Alignment with Implicit Reward Modeling. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*.

Lyu, Y.; Qin, X.; Du, X.; Zhao, N.; and Qiu, S. 2025. Multi-Path Reasoning for Multi-Hop Question Answering over Knowledge Graph. *Chinese Journal of Electronics*.

MAA. 2023. American Mathematics Competition 2023.

MAA. 2024. American Invitational Mathematics Examination 2024.

Muennighoff, N.; Yang, Z.; Shi, W.; Li, X. L.; Fei-Fei, L.; Hajishirzi, H.; Zettlemoyer, L.; Liang, P.; Candès, E.; and Hashimoto, T. 2025. s1: Simple test-time scaling. arXiv:2501.19393.

OpenAI; :; El-Kishky, A.; Wei, A.; Saraiva, A.; Minaiev, B.; Selsam, D.; Dohan, D.; Song, F.; Lightman, H.; Clavera, I.; Pachocki, J.; Tworek, J.; Kuhn, L.; Kaiser, L.; Chen, M.; Schwarzer, M.; Rohaninejad, M.; McAleese, N.; o3 contributors; Mürk, O.; Garg, R.; Shu, R.; Sidor, S.; Kosaraju, V.; and Zhou, W. 2025. Competitive Programming with Large Reasoning Models. arXiv:2502.06807.

OpenAI. 2024. OpenAI o1 System Card. arXiv:2412.16720.

Page, S. E.; et al. 2006. Path dependence. *Quarterly Journal of Political Science*.

Qin, Z.; He, Z.; Prakriya, N.; Cong, J.; and Sun, Y. 2025. Dynamic-width speculative beam decoding for llm inference. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*.

Qu, Y.; Jiang, Y.; Wang, B.; Mao, Y.; Wang, C.; Liu, C.; and Ji, X. 2025. Latent reward: Llm-empowered credit assignment in episodic reinforcement learning. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*.

Qwen; :; Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Li, C.; Liu, D.; Huang, F.; Wei, H.; Lin, H.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Lin, J.; Dang, K.; Lu, K.; Bao, K.; Yang, K.; Yu, L.; Li, M.; Xue, M.; Zhang, P.; Zhu, Q.; Men, R.; Lin, R.; Li, T.; Tang, T.; Xia, T.; Ren, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; and Qiu, Z. 2025. Qwen2.5 Technical Report. arXiv:2412.15115.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36: 53728–53741.

Rein, D.; Hou, B. L.; Stickland, A. C.; Petty, J.; Pang, R. Y.; Dirani, J.; Michael, J.; and Bowman, S. R. 2023. Gpqa: A graduate-level google-proof q&a benchmark. In *Proceedings of the 1st Conference on Language Modeling*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. arXiv:1707.06347.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv:2402.03300.

Sheng, G.; Zhang, C.; Ye, Z.; Wu, X.; Zhang, W.; Zhang, R.; Peng, Y.; Lin, H.; and Wu, C. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the 12th European Conference on Computer Systems*.

Sun, Y.; Zhang, Y.; Zhao, Z.; Wan, S.; Tao, D.; and Gong, C. 2025a. Fast-Slow-Thinking: Complex Task Solving with Large Language Models. *arXiv preprint arXiv:2504.08690*.

Sun, Y.; Zhao, Z.; Wan, S.; and Gong, C. 2025b. CortexDebate: Debating Sparsely and Equally for Multi-Agent Debate. In *Findings of the Association for Computational Linguistics: ACL 2025*.

Talmor, A.; Herzig, J.; Lourie, N.; and Berant, J. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North*. Association for Computational Linguistics.

Wang, A.; Song, L.; Tian, Y.; Peng, B.; Yu, D.; Mi, H.; Su, J.; and Yu, D. 2025a. LiteSearch: Efficient Tree Search with Dynamic Exploration Budget for Math Reasoning. In *Proceedings of the 39th AAAI Conference on Artificial Intelligence*.

Wang, S.; Yu, L.; Gao, C.; Zheng, C.; Liu, S.; Lu, R.; Dang, K.; Chen, X.; Yang, J.; Zhang, Z.; Liu, Y.; Yang, A.; Zhao, A.; Yue, Y.; Song, S.; Yu, B.; Huang, G.; and Lin, J. 2025b. Beyond the 80/20 Rule: High-Entropy Minority Tokens Drive Effective Reinforcement Learning for LLM Reasoning. arXiv:2506.01939.

Wang, S.; Yu, L.; Gao, C.; Zheng, C.; Liu, S.; Lu, R.; Dang, K.; Chen, X.; Yang, J.; Zhang, Z.; et al. 2025c. Beyond the 80/20 Rule: High-Entropy Minority Tokens Drive Effective Reinforcement Learning for LLM Reasoning. arXiv:2506.01939.

Wenjuan, L.; Chengxin, Z.; Hongbao, Z.; Bin, J.; and Baihang, L. 2025. RuleMaster+: LLM-Based Automated Rule Generation Framework for Intrusion Detection Systems. *Chinese Journal of Electronics*.

Xie, T.; Gao, Z.; Ren, Q.; Luo, H.; Hong, Y.; Dai, B.; Zhou, J.; Qiu, K.; Wu, Z.; and Luo, C. 2025. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. arXiv:2502.14768.

Xu, H.; Wu, X.; Wang, W.; Li, Z.; Zheng, D.; Chen, B.; Hu, Y.; Kang, S.; Ji, J.; Zhang, Y.; et al. 2025. RedStar: Does Scaling Long-CoT Data Unlock Better Slow-Reasoning Systems? arXiv:2501.11284.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; Zheng, C.; Liu, D.; Zhou, F.; Huang, F.; Hu, F.; Ge, H.; Wei, H.; Lin, H.; Tang, J.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Zhou, J.; Lin, J.; Dang, K.; Bao, K.; Yang, K.; Yu, L.; Deng, L.; Li, M.; Xue, M.; Li, M.; Zhang, P.; Wang, P.; Zhu, Q.; Men, R.; Gao, R.; Liu, S.; Luo, S.; Li, T.; Tang, T.; Yin, W.; Ren, X.; Wang, X.; Zhang, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Wang, Z.; Cui, Z.; Zhang, Z.; Zhou, Z.; and Qiu, Z. 2025. Qwen3 Technical Report. arXiv:2505.09388.

Yeo, E.; Tong, Y.; Niu, X.; Neubig, G.; and Yue, X. 2025. Demystifying Long Chain-of-Thought Reasoning in LLMs. In *ICLR 2025 Workshop on Deep Generative Model in Machine Learning: Theory, Principle and Efficacy*.

Yu, Q.; Zhang, Z.; Zhu, R.; Yuan, Y.; Zuo, X.; Yue, Y.; Fan, T.; Liu, G.; Liu, L.; Liu, X.; et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. arXiv:2503.14476.

Yu, W.; Jiang, Z.; Dong, Y.; and Feng, J. 2020. ReClor: A Reading Comprehension Dataset Requiring Logical Reasoning. In *Proceedings of the 8th International Conference on Learning Representations*.

Yue, Y.; Yuan, Y.; Yu, Q.; Zuo, X.; Zhu, R.; Xu, W.; Chen, J.; Wang, C.; Fan, T.; Du, Z.; et al. 2025. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. arXiv:2504.05118.

Zhang, D.; Zhoubian, S.; Hu, Z.; Yue, Y.; Dong, Y.; and Tang, J. 2024. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37: 64735–64772.

Zhang, J.; Lin, N.; Hou, L.; Feng, L.; and Li, J. 2025a. Adaptthink: Reasoning models can learn when to think. arXiv:2505.13417.

Zhang, Y.; Sun, Y.; Zhan, Y.; Tao, D.; Tao, D.; and Gong, C. 2025b. Large Language Models as an Indirect Reasoner: Contrapositive and Contradiction for Automated Reasoning. In *Proceedings of the 31st International Conference on Computational Linguistics*.

Zhao, X.; Kang, Z.; Feng, A.; Levine, S.; and Song, D. 2025. Learning to Reason without External Rewards. arXiv:2505.19590.

Zhao, Z.; Luo, L.; Pan, S.; Nguyen, Q. V. H.; and Gong, C. 2023. Towards few-shot inductive link prediction on knowledge graphs: A relational anonymous walk-guided neural process approach. In *Joint European conference on machine learning and knowledge discovery in databases*.

Zhao, Z.; Luo, L.; Pan, S.; Zhang, C.; and Gong, C. 2024. Graph Stochastic Neural Process for Inductive Few-shot Knowledge Graph Completion. *ACM Transactions on Intelligent Systems and Technology*.

Zheng, H.; Zhou, Y.; Bartoldson, B. R.; Kailkhura, B.; Lai, F.; Zhao, J.; and Chen, B. 2025. Act Only When It Pays: Efficient Reinforcement Learning for LLM Reasoning via Selective Rollouts. arXiv:2506.02177.