Incorporating Self-Rewriting into Large Language Model Reasoning Reinforcement

Jiashu Yao[1*], **Heyan Huang**[1], **Shuang Zeng**[3], **Chuwei Luo**[3], **Wangjie You**[3], **Jie Tang**[3], **Qingsong Liu**[3], **Yuhang Guo**[1†], **Yangyang Kang**[2,3†]

[1]Beijing Institute of Technology
[2]Zhejiang University
[3]ByteDance China
guoyuhang@bit.edu.cn, yangyangkang@bytedance.com

Abstract

Through reinforcement learning (RL) with outcome correctness rewards, large reasoning models (LRMs) with scaled inference computation have demonstrated substantial success on complex reasoning tasks. However, the one-sided reward, focused solely on final correctness, limits its ability to provide detailed supervision over internal reasoning process. This deficiency leads to suboptimal internal reasoning quality, manifesting as issues like over-thinking, under-thinking, redundant-thinking, and disordered-thinking. Inspired by the recent progress in LRM self-rewarding, we introduce self-rewriting framework, where a model rewrites its own reasoning texts, and subsequently learns from the rewritten reasoning to improve the internal thought process quality. For algorithm design, we propose a selective rewriting approach wherein only "simple" samples, defined by the model's consistent correctness, are rewritten, thereby preserving all original reward signals of GRPO. For practical implementation, we compile rewriting and vanilla generation within one single batch, maintaining the scalability of the RL algorithm and introducing only $\sim 10\%$ overhead. Extensive experiments on diverse tasks with different model sizes validate the effectiveness of self-rewriting. In terms of the accuracy-length tradeoff, the self-rewriting approach achieves improved accuracy (+0.6) with substantially shorter reasoning (-46%) even without explicit instructions in rewriting prompts to reduce reasoning length, outperforming existing strong baselines. In terms of internal reasoning quality, self-rewriting achieves significantly higher scores (+7.2) under the LLM-as-a-judge metric, successfully mitigating internal reasoning flaws.

1 Introduction

Using reinforcement learning (RL) for o1- and R1-like large reasoning model (LRM) post-training has demonstrated significant success in complex reasoning tasks with extended test-time computation (Shao et al. 2024; Guo et al. 2025; Qwen-Team 2025; Wu et al. 2025). The RL-based learning-to-reason paradigm provides trial and error reward signals focused on verifiable correctness (Xu et al. 2025). This approach offers training flexibility and ideally holds the potential to achieve superhuman reasoning intelligence (Wan et al. 2024; Han et al. 2025b,a; Silver et al. 2016, 2017).

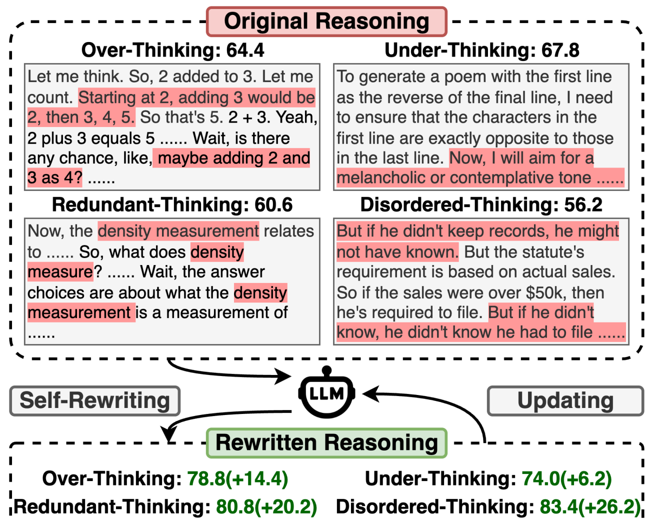

Figure 1: The reasoning of LRMs suffers from internal flaws. Over-thinking manifests as reasoning heavily over trivial or irrelevant aspects, under-thinking manifests as omissions and insufficient depth in reasoning, redundant-thinking manifests as repetition of essentially the same thoughts without bringing new ideas, and disordered-thinking manifests as jumping abruptly from one topic to another without coherence. Evaluated by LLM judges scores (the higher the better), rewriting significantly mitigates the flaws. Self-rewriting framework is to enable LRMs to learn from their own rewrites.

However, a careful examination of reasoning texts generated by current LRMs reveals numerous internal flaws, as exemplified in Figure 1. Specifically, we identify four common problem categories in LRM reasoning: (1) over-thinking, where models heavily define, calculate, or check irrelevant or trivial parts; (2) under-thinking, characterized by skipping or over-simplifying complex and relevant parts; (3) redundant-thinking, involving the repetition of essentially the same thoughts without introducing new ideas; and (4) disordered-thinking, which manifests as the interleaving of multiple thinking threads into a confusing mess. Collectively,

[*]The research and development work was carried out during Jiashu Yao's internship at ByteDance China.

[†]Corresponding authors.

these problems compromise the interpretability of the result, incur unnecessary reasoning costs due to the generation of meaningless texts, ultimately degrade the correctness of the final outcomes (Su et al. 2025).

Although recent researches on post-tuning LRMs for reasoning improvement are emerging, these efforts primarily focus solely on one single aspect of length control (Aggarwal and Welleck 2025; Yang et al. 2025; Chen et al. 2024), neglecting the complex reasoning internal flaws.

To address the need for fine-grained reasoning improvement, we draw inspiration from recent research on self-rewarding in LRMs (Ryu et al. 2024; Huang et al. 2023; Zhang et al. 2025c). This self-supervised learning paradigm involves models providing their own internal rewards during the RL process. Instead of generating numerical rewards, we propose self-rewriting. This novel approach instructs models to rewrite their own generated reasoning passages to enhance quality while preserving all core ideas. The models then learn from these rewritten versions of their reasoning. As illustrated in Figure 1, preliminary experiments demonstrate that self-rewriting can significantly mitigate the four aforementioned problems, as measured by LLM-as-a-judge metrics.

Specifically, we integrate a rewriting process into GRPO (Shao et al. 2024) and define a corresponding rewriting preference reward. Our approach employs selective rewriting, focusing only on "simple" queries where the model achieves 1.0 accuracy, while leaving other queries unchanged to minimize modifications to original GRPO algorithm. Furthermore, we designed an efficient implementation that introduces minimal computational overhead ($\sim 10\%$) when incorporating the rewriting process and preference.

Through comprehensive evaluation across diverse tasks and LRMs of varying sizes, we've made several key findings. Firstly, rewriting effectively serves as a reasoning length control method, even without explicit instructions in the rewriting prompt to reduce length. Compared to existing methods designed for efficient reasoning, our approach yields comparable or superior results in the trade-off between length and accuracy, answering correctly with fewer reasoning tokens. Furthermore, a detailed analysis validates the effectiveness of rewriting in generating reasoning of varying length and mitigating internal reasoning flaws, including over-thinking, under-thinking, redundant-thinking, and disordered-thinking.

Our main contributions are summarized as follows:

- We introduce the novel integration of reasoning rewriting into the LRM RL post-tuning framework to mitigate internal reasoning flaws. Our proposed self-rewriting framework enables models to learn from their rewritten reasoning, while maintaining the flexibility and scalability of the original GRPO algorithm.
- Extensive experiments across diverse tasks and multiple model sizes demonstrate that self-rewriting method exhibits strong length control capabilities, even without explicitly optimizing for length preference. It surpasses strong baselines in terms of the accuracy-length tradeoff.
- We conduct a fine-grained analysis of the rewritten texts. This validates that self-rewriting can generate more diverse response candidates of varying length, and successfully mitigate common reasoning flaws, including over-, under-, redundant-, and disordered-thinking.

2 Related Work

Reasoning Length Control Recent progress in LRMs (Guo et al. 2025; Qwen-Team 2025; Kimi-Team et al. 2025; Seed et al. 2025) has demonstrated the success of scaling test-time computation, where model performance on complex reasoning tasks consistently improves with the generation of more intermediate reasoning steps. However, a significant challenge for current LRMs is their tendency to generate an excessive number of reasoning tokens beyond necessary. This problem draws considerable research attention. One line of work explores modifying inference schemes to generate "soft thoughts", where each represents an entire reasoning passage (Hao et al. 2024; Shen et al. 2025; Zhang et al. 2025b). Another approach endeavors to address the issue within the text modality itself. These works generate responses of varying lengths by: (1) sampling multiple times (Aggarwal and Welleck 2025; Munkhbat et al. 2025; Su et al. 2025); (2) employing different prompts or thinking modes (Yang et al. 2025; Sun et al. 2025; Zhang et al. 2025a); or (3) explicitly truncating original responses into shorter ones (Chen et al. 2024; Qu et al. 2025; Dai, Yang, and Si 2025). The responses, now diverse in length, are then collected for fine-tuning, with a preference given to shorter and correct ones, ultimately aiming to generate concise and accurate reasoning paths.

Without explicit human-designed length preference, experiments show that self-rewriting method strikes a favorable accuracy-length tradeoff, outperforming strong baselines.

Reasoning Internal Quality Study Studying the internal quality of LRMs reasoning is more challenging than evaluating final accuracy or reasoning length, as the reasoning flaws cannot be directly formalized as an optimization objective. Nevertheless, reasoning quality, beyond just correctness and length, remains crucial for readability, interpretability, and overall performance (Su et al. 2025). Pioneering LRM research (Guo et al. 2025) leverages human-friendly cold-start fine-tuning to achieve readable formats and consistent language. More recently, Qi et al. (2025) proposes adapting reasoning to users' language to foster human trust and oversight. Another line of research (Lanham et al. 2023; Lyu et al. 2023) investigates reasoning consistency and aims to enhance its influence on the final result. Recent related work (Wang et al. 2025) identifies a prevalent switching-thought pattern in LRM reasoning, and proposes intervening in the inference process to encouraging reasoning in depth.

Unlike previous work that heuristically selects one internal aspect for analysis—such as readability, consistency, or switching-thought frequency—we aim to comprehensively improve reasoning by leveraging self-rewriting.

LLM Self-Rewarding The LLM self-rewarding mechanisms as alternatives to human-labeled rewards in fine-tuning is becoming increasingly promising, particularly as the capabilities of LLMs approach or surpass human expert performance on numerous tasks. Some research advocates for the use of generative reward models (Ryu et al. 2024; Mahan

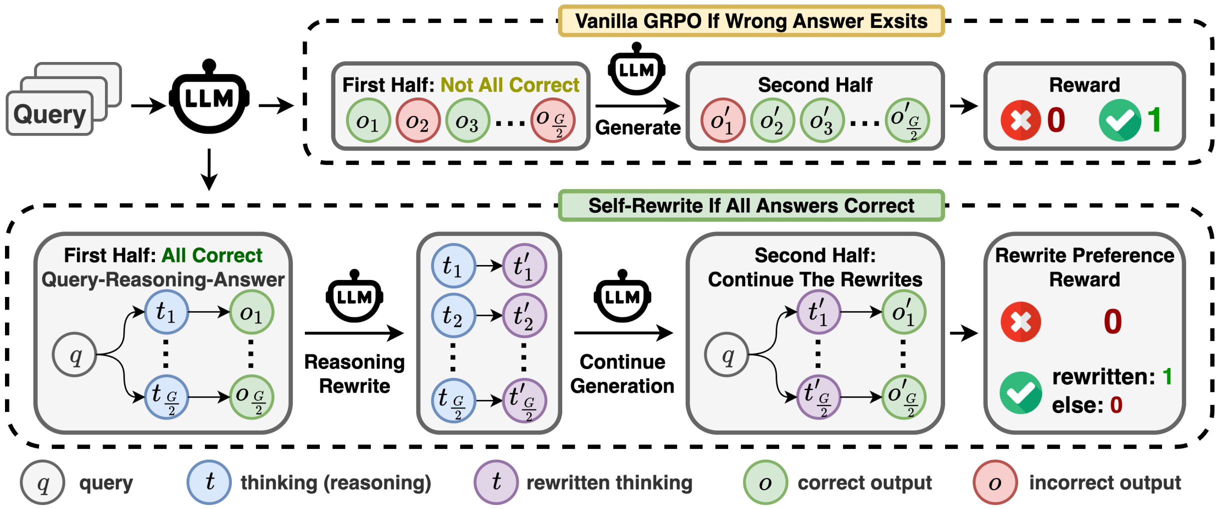

Figure 2: Self-rewriting framework. If any response for a given query is incorrect, the remaining half of the rollouts are sampled normally, and the final loss computation proceeds identically to GRPO. Conversely, if all initial responses for a query are correct, the model then rewrites and completes the reasoning texts, with the rewritten version receiving higher rewards.

et al. 2024; Yuan et al. 2024; Wu et al. 2024), while others propose employing majority voting or leveraging internal representation relations to estimate ground truth labels (Huang et al. 2023; Zuo et al. 2025; Zhang et al. 2025c). The LLM self-rewarding paradigm not only decreases the intensive demand for human labeling, but also achieves performance comparable to direct training with ground-truth data.

Our work extend this line of work, proposing novel self-rewriting for self-supervised LLM finetuning. Comparing to self-rewarding, self-rewriting adopts an generative rather than discrimitive approach to self-improvement, offering a more detailed and comprehensive guide for the actor model.

3 Method

Current LRM are supervised exclusively with verifable outcome rewards which focuses on the correctness of the results during RL training, lacking an explicit mechanism for internal quality improvement that comprehensively addressing reasoning flaws. These flaws compromise (1) the reasoning interpretability and oversight, (2) the reasoning efficiency, and (3) the final performance.

Inspired by recent advancements in self-rewarding mechanisms, we leverage the general language understanding and generation capabilities of LLMs to instruct models to rewrite their own reasoning passages. To enable learning from this rewritten reasoning while retaining the original performance, we integrate the selective rewriting process as a component into GRPO. Combining these elements, we propose the self-rewriting framework and present its efficient implementation.

Reasoning Rewriting Reasoning rewriting aims to change the features of the original reasoning and improve the overall quality. Addressing specific application scenarios, the rewriting instruction can be quite flexible, changing the reasoning style and so on. While in this work, we are to examine the basic principle of the method, thus only adopting a minimum rewriting instruction which focuses on general quality improvement and not involves special requirement. The general prompt for rewriting is shown in the text box below.

> You are a skilled editor tasked with improving a given thinking passage. Your goal is to refine the passage to enhance its overall quality, making it more organized, coherent, and accurate. Your output should be a rewritten version of the original thinking passage. The rewritten version should maintain the core ideas and essence of the original while significantly improving its presentation and impact. Please provide only the rewritten thinking passage, without any additional explanations or context.

Selective Rewriting By incorporating rewriting into the RL process, we aim to enable the LLM to learn from its own rewrites, thereby improving presentation and eliminating redundancy while retaining high reasoning effectiveness. Inspired by recent work that filters samples with 1.0 accuracy (Yu et al. 2025), we propose to apply selective rewriting, as illustrated in Figure 2. This approach exclusively rewrites those samples that are already completely correct. Formally,

$$\pi_\theta^{SR}(T, O|q, \{t_i, o_i\}_{i=1}^{\frac{G}{2}}) = \begin{cases} \pi_\theta(T|\{t_i\}_{i=1}^{\frac{G}{2}})\ \pi_\theta(O|q, T) & \text{if } \{o_i\}_{i=1}^{\frac{G}{2}} \text{ all correct} \\ \pi_\theta(T, O|q) & \text{else} \end{cases} \tag{1}$$

where q, t, o denotes query, reasoning text, and final answer respectively. If and only if the first half group of responses are all correct, selective rewriting strategy rewrites the reasoning, consequently concatenates the original query and rewritten reasoning for continue generation of final answers. Otherwise, the following half group are sampled normally.

Selective rewriting strategy offers two key advantages. Firstly, it minimally interrupts current optimization methods, utilizing only what would otherwise be useless samples in the original GRPO process. Secondly, by focusing on the simplest queries that the LLM has already mastered, it allows the model to practice generating concise and well-presented thoughts on easy problems, while still engaging in normal sampling for harder ones, preserving the improvement space for complex problem-solving.

Optimization Self-rewriting framework gives higher rewards to rewritten samples if all responses are correct; otherwise, it gives higher rewards to correct samples. Formally, given the raw correctness reward $\mathbf{r}$, the reward for the i-th sample in a group is defined as

$$\hat{r}_i = \begin{cases} r_i & \text{if } \mathbf{r} \neq \mathbf{1}, \\ 1 & \text{if } \mathbf{r} = \mathbf{1} \wedge (i\text{-th sample is rewritten}), \\ 0 & \text{else.} \end{cases} \quad (2)$$

Calculate the advantage $\hat{A}_{i,t}$ based on $\hat{\mathbf{r}}$ (details are discussed in Appendix A), the policy model is optimized by maximizing the objective:

$$\mathcal{J}_\theta = \mathbb{E}\Big[q \sim P(Q), \{t, o\}_{i=1}^{\frac{G}{2}} \sim \pi_{\theta_{old}}(\cdot|q), \\ \{t, o\}_{i=\frac{G}{2}+1}^{G} \sim \pi_{\theta_{old}}^{SR}(\cdot|q, \{t_i, o_i\}_{i=1}^{\frac{G}{2}})\Big] \quad (3) \\ \frac{\sum_{i=1}^{G}}{G} \frac{\sum_{t=1}^{|o_i|}}{|o_i|} \Big\{ \min[\frac{\pi_\theta}{\pi_{\theta_{\text{old}}}} \hat{A}_{i,t}, \text{clip}(\frac{\pi_\theta}{\pi_{\theta_{\text{old}}}}, 1 \pm \epsilon) \hat{A}_{i,t}] \Big\}.$$

Overall Algorithm The pseudocode of self-rewriting is shown in Algorithm 1.

Algorithm 1: SELF-REWRITING

Input: query set Q, model $\mathcal{M}$, verifier $\mathcal{R}$, group size G
Output: adapted model $\mathcal{M}$

for q in Q **do**
  $\mathbf{t}_{1:\frac{G}{2}}, \mathbf{o}_{1:\frac{G}{2}} \leftarrow \mathcal{M}(q)$ // generate $\frac{G}{2}$ responses
  $\mathbf{r}_{1:\frac{G}{2}} \leftarrow \mathcal{R}(q, \mathbf{o}_{1:\frac{G}{2}})$ // verify first half batch
  if $\mathbf{r}_{1:\frac{G}{2}} = \mathbf{1}$ **then**
    $\mathbf{t}_{\frac{G}{2}+1:G} \leftarrow \mathcal{M}(\mathbf{t}_{1:\frac{G}{2}})$ // rewrite reasoning
    $\mathbf{o}_{\frac{G}{2}+1:G} \leftarrow \mathcal{M}(q, \mathbf{t}_{\frac{G}{2}+1:G})$ // continue generation
  else
    $\mathbf{t}_{\frac{G}{2}+1:G}, \mathbf{o}_{\frac{G}{2}+1:G} \leftarrow \mathcal{M}(q)$ // vanilla generation
  end if
  $\mathbf{r}_{1:G} \leftarrow \mathcal{R}(q, \mathbf{o}_{1:G})$
  $\hat{\mathbf{r}}_{1:G} \leftarrow$ Equ2$(\mathbf{r}_{1:G})$ // reward with Equation 2
  $\mathcal{M} \leftarrow step(\mathcal{M}, q, \mathbf{t}, \mathbf{o}, \hat{\mathbf{r}})$ // GRPO stepping
end for
return $\mathcal{M}$

Efficient Implementation Self-rewriting complicates the RL process by incorporating a selective rewriting after generating the first half batch. To ensure the added complexity does not lead to significantly higher computation consumption, we design a practically efficient implementation for the algorithm. As is shown in Figure 3, after the generation of the first half batch, we compile the vanilla generation samples from not-all-correct queries and the rewriting samples from all-correct queries into one batch for joint inference. Subsequently, the rewritten reasoning segments are separately compiled into another batch for continued generation. Note that the continuation only generates outputs after </think> token, which usually takes a very small portion of the overall response. Compared to vanilla GRPO, the implementation increases the time consumption by only about 10%, which is acceptable given that our method is primarily intended for post-tuning rather than large-scale pre-training.

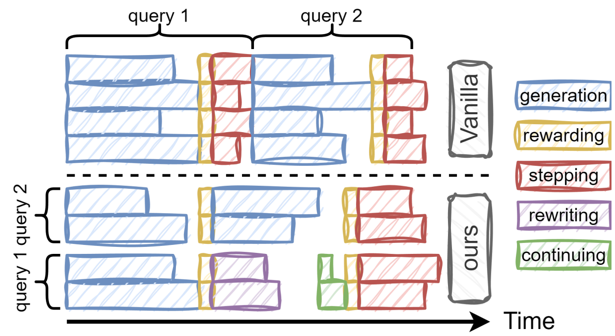

Figure 3: The illustration of computation process for vanilla GRPO (top) and self-rewriting (bottom), when there are 4 computation units, 2 queries in a batch, and 4 rollouts for each query. By compiling generation and rewriting into one batch, the complexity introduced by selective rewriting only leads to time consumption increment of about 10%.

4 Experiments

4.1 Setups

We list main experimental setups here, and detailed experimental information are shown in Appendix B and C.

Datasets For training datasets, we randomly sample 10K pieces from DeepMath-103K (He et al. 2025) following previous work (Dai, Yang, and Si 2025). For comprehensive evaluation, we conduct test on four diverse types of tasks, including math reasoning MATH-500 (Lightman et al. 2023), science reasoning GPQA-Diamond (Rein et al. 2024), logic reasoning ARC-Challenging (Clark et al. 2018), and knowledge reasoning MMLU-Pro (Wang et al. 2024).

Metrics Following previous work, we firstly measure the tradeoff between accuracy and length. Specifically, we evaluate the average $pass@1$ rate and token numbers across 4 sampled runs with temperature 0.6 and max length 32K. In addition, we are also interested in the internal quality of reasoning texts, i.e., the over-thinking, under-thinking, redundant-thinking, and disordered-thinking problems. However, LLM reasoning texts are often very long and require professional level knowledge to understand, thus very difficult to evaluate. For efficient and effective evaluation, we use stronger LLMs (DeepSeek-V3, others in Appendix D) instructed to judge the reasoning text according to the four aspects (prompts shown in Appendix B), and score in range of 1 to 5. We scale the averaged results into 100.

Models	MATH-500			GPQA-Diamond			ARC-Challenge			MMLU-Pro			Average		
	Acc	Len	Jdg	Acc	Len	Jdg	Acc	Len	Jdg	Acc	Len	Jdg	Acc	Len	Jdg
Qwen3-1.7B															
Original	86.0	5267	71.5	37.9	7999	51.0	87.0	798	74.6	56.6	3354	64.3	66.9(+0.0)	4355(+0%)	65.4(+0.0)
GRPO	87.0	4695	71.2	37.9	6960	50.7	86.1	717	77.1	56.6	3105	66.6	**66.9(+0.0)**	3869(-11%)	66.4(+1.0)
LenPen1	83.8	3794	77.5	33.8	5637	52.5	87.3	591	81.4	56.4	2381	69.4	65.3(-1.6)	3101(-29%)	70.2(+4.8)
LenPen2	83.7	3893	74.4	34.2	5814	52.7	87.1	597	80.8	55.9	2527	68.6	65.2(-1.7)	3209(-27%)	69.1(+3.7)
ShortBetter	86.0	3714	75.4	37.1	5947	53.0	86.1	607	82.1	56.4	2577	68.1	66.4(-0.5)	3211(-26%)	69.7(+4.3)
LPO	83.4	3854	73.0	36.0	4974	53.1	86.0	641	80.6	55.5	2386	65.2	65.2(-1.7)	2964(-32%)	68.0(+2.6)
TOPS	80.7	3855	74.3	36.2	4632	51.4	86.3	580	80.9	55.3	2238	68.1	64.6(-2.3)	**2826(-35%)**	68.7(+3.3)
Rewrite	86.0	3820	79.6	36.9	4823	55.5	86.3	503	84.9	55.7	2195	74.0	66.2(-0.7)	2835(-35%)	**73.5(+8.1)**
Qwen3-4B															
Original	89.6	4544	73.4	52.2	7373	54.5	92.7	742	75.4	69.9	3440	64.9	76.1(+0.0)	4025(+0%)	67.1(+0.0)
GRPO	90.0	3910	74.7	52.5	7057	54.3	92.4	692	77.7	68.1	3063	67.8	75.8(-0.3)	3681(-9%)	68.6(+1.5)
LenPen1	89.2	2634	81.4	54.0	4650	57.8	92.5	480	83.2	69.4	2041	73.2	76.3(+0.2)	2451(-39%)	73.9(+6.8)
LenPen2	89.4	2607	81.9	52.0	4966	57.8	92.9	491	83.0	69.6	2091	70.3	76.0(-0.1)	2539(-37%)	73.3(+6.2)
ShortBetter	89.2	2480	84.6	49.2	4904	57.3	92.8	499	82.4	67.8	2196	72.6	74.8(-1.3)	2520(-37%)	74.2(+7.1)
LPO	89.4	3142	79.6	48.8	4602	57.3	92.1	598	79.4	66.8	2277	69.0	74.3(-1.8)	2655(-34%)	71.3(+4.2)
TOPS	88.6	2929	85.2	46.8	3772	55.3	92.2	419	81.0	66.6	1694	73.0	73.6(-2.5)	**2204(-45%)**	73.6(+6.5)
Rewrite	89.4	2005	88.8	54.6	4647	58.6	93.0	468	85.8	68.9	1747	76.2	**76.5(+0.4)**	2217(-45%)	**77.4(+10.3)**
Qwen3-8B															
Original	90.2	4663	78.6	55.1	8063	58.5	93.5	764	82.1	74.1	3482	69.3	78.2(+0.0)	4243(+0%)	72.1(+0.0)
GRPO	89.6	4446	78.6	53.0	7178	56.9	93.9	728	82.5	76.4	3194	69.8	78.2(+0.0)	3887(-8%)	72.0(-0.1)
LenPen1	89.0	3034	83.6	55.6	5371	58.8	93.9	548	85.4	74.3	2353	72.1	78.2(+0.0)	2827(-33%)	75.0(+2.9)
LenPen2	89.2	2938	82.9	56.6	5929	59.0	94.0	560	84.5	74.3	2380	73.5	78.5(+0.3)	2952(-30%)	75.0(+2.9)
ShortBetter	90.4	3001	83.7	53.3	5754	58.0	93.7	560	86.6	74.5	2402	72.9	78.0(-0.2)	2929(-31%)	75.3(+3.2)
LPO	89.6	3648	80.9	52.3	5666	56.7	94.1	625	82.3	73.8	2698	70.6	77.5(-0.7)	3159(-26%)	72.6(+0.5)
TOPS	89.3	2796	83.4	53.8	4505	56.7	94.2	474	86.4	73.9	2048	72.4	77.8(-0.4)	2456(-42%)	74.7(+2.6)
Rewrite	89.2	2490	90.0	57.1	4018	61.2	94.3	463	89.5	74.4	2202	76.6	**78.8(+0.6)**	**2293(-46%)**	**79.3(+7.2)**

Table 1: Main experimental results. Best average results for each model are bolded.

Models We conduct experiments with well-established LRMs with varied parameter sizes, including Qwen3-1.7B, Qwen3-4B, and Qwen3-8B.

Baselines (1) GRPO (Shao et al. 2024) reinforces correctness rewards. (2) Length Penalty is a simple but effectively method widely used by previous work (Kimi-Team et al. 2025; Hiroshi, Taiki, and Yuichi 2025; Arora and Zanette 2025), which heuristically adopts length rewards as supplements to correctness rewards favoring shorter responses. We implements two types of length penalty proposed by Kimi-Team et al. (2025) and Arora and Zanette (2025), referring them as LP1 and LP2 respectively. (3) ShorterBetter (Yi, Wang, and Li 2025) prefers responses whose length close to the shortest correct ones, instead of simply shorter ones. (4) LPO (Su et al. 2025) leverage offline-RL to prefer shorter responses over longer ones. (5) TOPS (Yang et al. 2025) applies reasoning effort-conditioned generation for multiple responses with diverse conciseness level, and adopts offline-RL preferring the shortest correct reasoning.

Implementation For result reliability and robustness, we adopt commonly-used hyper-parameters without extensive search. All methods are trained for 1 epoch with batch size 256. For online RL, we use GRPO with learning rate as $3e-6$ and rollout size as 8. For offline RL, we use SimPO (Meng, Xia, and Chen 2024) with learning rate as $1e-6$, $\beta = 2.0$ and $\gamma = 0.3$. Other details are listed in Appendix C.

4.2 Results

The main results are shown in Table 1.

Accuracy-Length Tradeoff All methods experimented manage to shorten reasoning length, including vanilla GRPO with a cutoff length 12K. Among them, our proposed self-rewriting method strikes a good tradeoff between accuracy and reasoning length, outperforming existing strong baselines. Specifically, with in the same training load, self-rewriting is able to generate more concise reasoning leading to more correct final answers, due to the strength of LLM rewriting supervision over previous length preference.

LLM Judge Scores LLM judge scores assess issues such as over-thinking, under-thinking, redundant-thinking, and disordered-thinking in reasoning texts by leveraging the general language understanding capabilities of powerful LLMs. The results reveal that stronger models yield higher judge scores, while length control methods can further enhance the internal quality of reasoning. Notably, self-rewriting achieves significantly higher LLM judge scores compared to all baseline methods, demonstrating its capacity to improve internal reasoning quality and alleviate reasoning flaws. These substantially higher judge scores partially explain why self-rewriting achieves a favorable accuracy-length tradeoff.

Online versus Offline Our comparison between online RL methods (LenPen1, LenPen2, ShorterBetter, and our proposed self-rewriting) and offline RL methods (LPO and TOPS) reveals two key distinctions. First, regarding final accuracy and LLM judge scores, online methods generally outperform offline approaches even when offline methods like TOPS also incorporate correctness preferences. This demonstrates the importance of online generation for maintaining performance, as models require dynamic reward signal calibration alongside parameter updates, rather than relying on static preference datasets constructed before fine-tuning begins. Second, online RL methods demonstrate more proportional length control than offline approaches on out-of-domain tasks. When trained on math reasoning data, offline methods tend to over-truncate GPQA reasoning which requires more detailed explanations, while under-truncating simpler ARC tasks. In contrast, online methods adjust lengths more appropriately for all tasks. In summary, for length control scenarios requiring both accuracy preservation and generalization beyond training data, online methods, particularly our proposed self-rewriting, offer superior performance.

4.3 Ablations

We conduct further ablation experiments to answer two key questions. (1) Does our proposed self-rewriting method consistently outperform its baselines across different length compression scales? (2) How does selective rewriting policy influences the final results?

Training Data Scale To compare our self-rewriting method with its baselines across different lengths, we continue training the online RL methods for an additional 10K samples (20K samples in total). We then evaluate four checkpoints corresponding to 5K, 10K, 15K, and 20K training samples for each method. The results are presented in Figure 4.

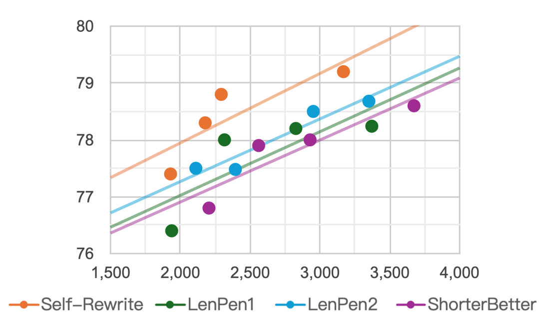

Figure 4: Results of four checkpoints for each methods on Qwen3-8B. X-axis refers to length and y-axis refers to average accuracy across four tasks.

The results show that, with more training, length control methods progressively shorten reasoning length at the cost of compromising accuracy. Among them, self-rewriting consistently outperforms other online RL baselines for accuracy under different length budgets.

Selective Rewriting We propose selective rewriting to let the model exclusively rewrite "simple" samples for better presentation, while explore for correctness for other ones. To validate the effectiveness of selective rewriting, we compare it with a vanilla rewriting policy, where at each step, $n\%$ of samples are randomly chosen for rewriting. As the results in Table 2 show, the performance is degraded without selective rewriting process.

Models	Acc	Len	Jdg
Original	78.2(+0.0)	4243(+0%)	72.1(+0.0)
GRPO	78.2(+0.0)	3887(-8%)	72.0(-0.1)
Self-Rewriting	**78.8(+0.6)**	2293(-46%)	**79.3(+7.2)**
w/o SR(50%)	77.9(-0.3)	2691(-37%)	78.8(+6.7)
w/o SR(100%)	77.8(-0.4)	**2120(-50%)**	78.6(+6.5)

Table 2: Ablation experimental results on Qwen3-8B. All results are averaged on four tasks. "SR" denotes selective rewriting, and "without SR ($n\%$)" refers to the process that randomly select $n\%$ samples to rewrite, instead of choosing correct ones as in selective rewriting. Best results are bolded.

5 Analysis

In previous experiments, we have validated the high effectiveness of our proposed self-rewriting over previous length control methods in terms of accuracy-length tradeoff, and we attribute such superiority to the high reasoning text understanding and rewriting capability of LLMs. Specifically, we speculate that LLMs are able to rewrite the reasoning texts generated by their own for higher internal reasoning quality, mitigating the issues including over-thinking, under-thinking, redundant-thinking, and disordered-thinking.

While our hypothesis that self-rewriting can improve reasoning internal quality has been validated through the LLM-as-a-judge methods of previous experiments as shown in Table 1, we want to have a deeper understanding about how rewriting changes the character of reasoning texts in details. In this section, we conduct further analyses to offer more insights about how rewriting improve the reasoning texts.

Before presenting detailed results, we emphasize that all rewriting analyses are based on the neutral prompts described in Section 3. Our tested rewriting prompt uses only basic instructions to improve reasoning quality generically, without revealing our specific evaluation criteria (over-thinking, etc.) or explicitly requesting conciseness. While application-specific scenarios could employ more targeted prompts, we intentionally use minimal instructions to enable fundamental evaluation of the core methodology of self-rewriting.

5.1 Length Ratio Distribution

Previous reasoning length control approaches typically incorporate length preference as supplements to correctness rewards, while self-rewriting framework rewrites reasoning texts and prefers the rewritten responses in rewarding. We first examine the distribution of length ratios between preferred and rejected responses under self-rewriting approach and previous length preference approaches.

In practice, we analyze the first 20 steps (corresponding to 5K samples) with Qwen3-8B self-rewriting training, and count the ratio between preferred responses (rewritten ones) and rejected responses (original ones). We compare the length ratio distribution with that of previous length control approaches, which denotes the ratio between the shortest correct responses and the other correct responses.

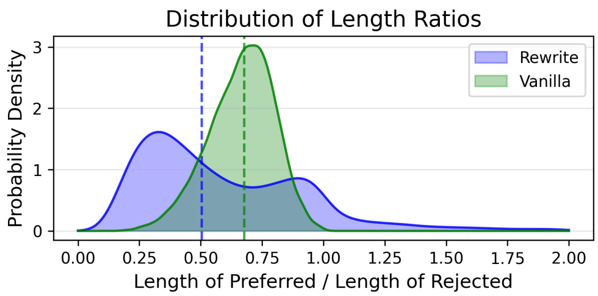

Figure 5: The length ratio distribution of Qwen3-8B online RL training dataset between preferred and other responses. Preferred responses refer to rewritten ones for self-rewriting, and shortest correct ones for vanilla length control approaches. The vertical dashed lines denote median values. The probability density function shown in the figure is obtained using Gaussian kernel density estimation.

The results in Figure 5 reveal three key insights. First, self-rewriting in general adopts a more aggressive length reduction strategy, with a median length ratio of $\sim$0.5 compared to $\sim$0.7 for vanilla sampling methods. This explains Table 1's findings where self-rewriting achieves shorter responses with equivalent training. Second, self-rewriting exhibits significantly greater variance in length ratios, indicating its greater ability to generate more diverse response candidates in terms of length. Notably, $\sim$10% of rewritten samples become longer than originals, which is an outcome impossible with vanilla length preferences. Finally, self-rewriting displays a bimodal distribution, rather than the unimodal distribution pattern of vanilla length preference, suggesting LLMs dynamically adapt conciseness based on problem characteristics rather than applying uniform length reduction.

5.2 Fine-Grained LLM Judge Scores

Our discussion has revealed the internal reasoning flaws in current LRMs as shown in Figure 1, manifesting as over-thinking, under-thinking, redundant-thinking, and disordered-thinking. Moreover, through rigorous LLM-as-a-judge evaluation with powerful model evaluators, we have demonstrated the self-rewriting's significant mitigation of these issues, as shown in Table 1. We now conduct a fine-grained examination of how rewriting enhances reasoning quality.

Using the same judge model and prompts from our main experiments, we analyze fine-grained scores for each reasoning flaw category: over-thinking, under-thinking, redundant-thinking, and disordered-thinking. Our primary evaluation focuses on the DeepMath training data, which is also used in our main experiments, to understand how self-rewriting improves reasoning quality. For generalizability, we conduct additional testing on the MMLU training set. 1K queries are sample from both datasets.

Models	Over	Under	Redundant	Disordered
DeepMath				
Original	76.3	82.0	67.9	61.6
Rewritten	84.0	87.3	82.1	77.1
MMLU				
Original	64.4	67.8	60.6	56.2
Rewritten	78.8	74.0	80.8	83.4

Table 3: LLM-as-a-judge scores of reasoning texts on four dimensions, including over-thinking, under-thinking, redundant-thinking, and disordered-thinking. The original the rewritten reasoning texts are all generated by Qwen3-8B.

The results in Table 3 reveal consistent patterns across both datasets. Redundant-thinking and disordered-thinking show significant improvement evidenced by higher scores, while over-thinking and under-thinking demonstrate more modest gains. This aligns with expectations, as LLM reflection can more easily identify repetitive content and incoherent topic transitions compared to determining optimal thought depth and coverage. Notably, direct rewriting yields greater LLM judge score improvements than fine-tuning with our self-rewriting framework. This suggests the online RL framework's correctness scores prevent overfitting to rewritten outputs, maintaining (or even improving) accuracy despite smaller judge score gains, demonstrating an effective balance.

6 Conclusion

This paper proposes self-rewriting, an RL reasoning reinforcement framework that incorporated with model-generated rewrites. Self-rewriting framework employs selective rewriting to minimize disruption to the vanilla RL process, focusing exclusively on simple queries while favoring their rewritten reasoning texts. Our implementation optimizes efficiency by compiling continuous generation and rewriting into a single batch operation, obtaining a computational overhead as low as $\sim$ 10%. Extensive experiments across diverse tasks and model sizes validate the superior reasoning improvement ability of self-rewriting over strong baselines, particularly in balancing accuracy and length while achieving higher reasoning quality according to LLM judge metrics.

Self-rewriting extends current line of self-rewarding approaches for reasoning improvement, demonstrating that LLMs can generate concise high-quality supervision through rewriting their own reasoning texts. The framework's flexibility allows for targeted rewriting by modifying prompt instructions, enabling the generation of texts with specific features tailored to particular applications. Tthis work focuses on general rewriting instructions, and leaves the exploration of specialized targeted rewriting for future research.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. U21B2009).

References

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

Arora, D.; and Zanette, A. 2025. Training language models to reason efficiently. *arXiv preprint arXiv:2502.04463*.

Chen, X.; Xu, J.; Liang, T.; He, Z.; Pang, J.; Yu, D.; Song, L.; Liu, Q.; Zhou, M.; Zhang, Z.; et al. 2024. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Dai, M.; Yang, C.; and Si, Q. 2025. S-GRPO: Early Exit via Reinforcement Learning in Reasoning Models. *arXiv preprint arXiv:2505.07686*.

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 preprint arXiv:2501.12948*.

Han, S.; Pari, J.; Gershman, S. J.; and Agrawal, P. 2025a. General reasoning requires learning to reason from the get-go. *arXiv preprint arXiv:2502.19402*.

Han, S.; Pari, J.; Gershman, S. J.; and Agrawal, P. 2025b. Position: General Intelligence Requires Reward-based Pretraining. In *Forty-second International Conference on Machine Learning Position Paper Track*.

Hao, S.; Sukhbaatar, S.; Su, D.; Li, X.; Hu, Z.; Weston, J.; and Tian, Y. 2024. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*.

He, Z.; Liang, T.; Xu, J.; Liu, Q.; Chen, X.; Wang, Y.; Song, L.; Yu, D.; Liang, Z.; Wang, W.; Zhang, Z.; Wang, R.; Tu, Z.; Mi, H.; and Yu, D. 2025. DeepMath-103K: A Large-Scale, Challenging, Decontaminated, and Verifiable Mathematical Dataset for Advancing Reasoning. *arXiv preprint arXiv:2504.11456*.

Hiroshi, Y.; Taiki, Y.; and Yuichi, I. 2025. A Practical Two-Stage Recipe for Mathematical LLMs: Maximizing Accuracy with SFT and Efficiency with Reinforcement Learning. *arXiv preprint arXiv:2507.08267*.

Huang, J.; Gu, S. S.; Hou, L.; Wu, Y.; Wang, X.; Yu, H.; and Han, J. 2023. Large Language Models Can Self-Improve. In *2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023*, 1051–1068. Association for Computational Linguistics (ACL).

Kimi-Team; Du, A.; Gao, B.; Xing, B.; Jiang, C.; Chen, C.; Li, C.; Xiao, C.; Du, C.; Liao, C.; et al. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Lanham, T.; Chen, A.; Radhakrishnan, A.; Steiner, B.; Denison, C.; Hernandez, D.; Li, D.; Durmus, E.; Hubinger, E.; Kernion, J.; et al. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*.

Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Lyu, Q.; Havaldar, S.; Stein, A.; Zhang, L.; Rao, D.; Wong, E.; Apidianaki, M.; and Callison-Burch, C. 2023. Faithful Chain-of-Thought Reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, 305–329.

Mahan, D.; Van Phung, D.; Rafailov, R.; Blagden, C.; Lile, N.; Castricato, L.; Fränken, J.-P.; Finn, C.; and Albalak, A. 2024. Generative reward models. *arXiv preprint arXiv:2410.12832*.

Meng, Y.; Xia, M.; and Chen, D. 2024. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37: 124198–124235.

Munkhbat, T.; Ho, N.; Kim, S. H.; Yang, Y.; Kim, Y.; and Yun, S.-Y. 2025. Self-training elicits concise reasoning in large language models. *arXiv preprint arXiv:2502.20122*.

Qi, J.; Chen, S.; Xiong, Z.; Fernández, R.; Bitterman, D. S.; and Bisazza, A. 2025. When Models Reason in Your Language: Controlling Thinking Trace Language Comes at the Cost of Accuracy. *arXiv e-prints*, arXiv–2505.

Qu, Y.; Yang, M. Y.; Setlur, A.; Tunstall, L.; Beeching, E. E.; Salakhutdinov, R.; and Kumar, A. 2025. Optimizing Test-Time Compute via Meta Reinforcement Finetuning. In *Forty-second International Conference on Machine Learning*.

Qwen-Team. 2025. QwQ-32B: Embracing the Power of Reinforcement Learning.

Rein, D.; Hou, B. L.; Stickland, A. C.; Petty, J.; Pang, R. Y.; Dirani, J.; Michael, J.; and Bowman, S. R. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Ryu, S.; Do, H.; Kim, Y.; Lee, G.; and Ok, J. 2024. Multi-Dimensional Optimization for Text Summarization via Reinforcement Learning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 5858–5871.

Seed, B.; Chen, J.; Fan, T.; Liu, X.; Liu, L.; Lin, Z.; Wang, M.; Wang, C.; Wei, X.; Xu, W.; et al. 2025. Seed1. 5-thinking: Advancing superb reasoning models with reinforcement learning. *arXiv preprint arXiv:2504.13914*.

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

Shen, Z.; Yan, H.; Zhang, L.; Hu, Z.; Du, Y.; and He, Y. 2025. Codi: Compressing chain-of-thought into continuous space via self-distillation. *arXiv preprint arXiv:2502.21074*.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of go without human knowledge. *nature*, 550(7676): 354–359.

Su, J.; Healey, J.; Nakov, P.; and Cardie, C. 2025. Between underthinking and overthinking: An empirical study of reasoning length and correctness in llms. *arXiv preprint arXiv:2505.00127*.

Sun, Y.; Wang, H.; Li, J.; Liu, J.; Li, X.; Wen, H.; Yuan, Y.; Zheng, H.; Liang, Y.; Li, Y.; et al. 2025. An Empirical Study of LLM Reasoning Ability Under Strict Output Length Constraint. *arXiv preprint arXiv:2504.14350*.

Wan, Z.; Feng, X.; Wen, M.; McAleer, S.; Wen, Y.; Zhang, W.; and Wang, J. 2024. AlphaZero-Like Tree-Search can Guide Large Language Model Decoding and Training. In *Proceedings of Machine Learning Research*, volume 235, 49890–49920. NeuriPS.

Wang, Y.; Liu, Q.; Xu, J.; Liang, T.; Chen, X.; He, Z.; Song, L.; Yu, D.; Li, J.; Zhang, Z.; et al. 2025. Thoughts are all over the place: On the underthinking of o1-like llms. *arXiv preprint arXiv:2501.18585*.

Wang, Y.; Ma, X.; Zhang, G.; Ni, Y.; Chandra, A.; Guo, S.; Ren, W.; Arulraj, A.; He, X.; Jiang, Z.; et al. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37: 95266–95290.

Wu, T.; Yuan, W.; Golovneva, O.; Xu, J.; Tian, Y.; Jiao, J.; Weston, J.; and Sukhbaatar, S. 2024. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. *arXiv preprint arXiv:2407.19594*.

Wu, Y.; Sun, Z.; Li, S.; Welleck, S.; and Yang, Y. 2025. Inference scaling laws: An empirical analysis of compute-optimal inference for LLM problem-solving. In *The Thirteenth International Conference on Learning Representations*.

Xu, F.; Hao, Q.; Zong, Z.; Wang, J.; Zhang, Y.; Wang, J.; Lan, X.; Gong, J.; Ouyang, T.; Meng, F.; et al. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*.

Yang, W.; Ma, S.; Lin, Y.; and Wei, F. 2025. Towards thinking-optimal scaling of test-time compute for llm reasoning. *arXiv preprint arXiv:2502.18080*.

Yi, J.; Wang, J.; and Li, S. 2025. Shorterbetter: Guiding reasoning models to find optimal inference length for efficient reasoning. *arXiv preprint arXiv:2504.21370*.

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

Yuan, W.; Pang, R. Y.; Cho, K.; Sukhbaatar, S.; Xu, J.; and Weston, J. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 3.

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

Zhang, J.; Zhu, Y.; Sun, M.; Luo, Y.; Qiao, S.; Du, L.; Zheng, D.; Chen, H.; and Zhang, N. 2025b. Lightthinker: Thinking step-by-step compression. *arXiv preprint arXiv:2502.15589*.

Zhang, Q.; Wu, H.; Zhang, C.; Zhao, P.; and Bian, Y. 2025c. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*.

Zuo, Y.; Zhang, K.; Sheng, L.; Qu, S.; Cui, G.; Zhu, X.; Li, H.; Zhang, Y.; Long, X.; Hua, E.; et al. 2025. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*.