LC-R1: Optimizing Length Compression in Large Reasoning Model

Anonymous EMNLP submission

Abstract

Large Reasoning Models (LRMs) have made great progress in complex reasoning tasks by being trained to generate step-by-step thinking paths. However, the length of these models' outputs also increases drastically with unnecessary reasoning chains—a phenomenon termed *"overthinking"*—especially when solving simple problems with clear solution paths. This paper introduces three principles for efficient reasoning: Simplicity (minimizing redundant content), Sufficiency (ensuring critical reasoning steps are retained), and Accuracy (arriving at correct answers). Motivated by them, we introduce LC-R1, a reinforcement learning (RL) algorithm introducing a novel collaboration of length reward and a compress reward/penalty, in addition to the accuracy reward. Hence, it encourages compression that can preserve the accuracy and completeness of the thinking process. Extensive experiments across five mathematical reasoning benchmarks with Distill-Qwen-1.5B/7B as base models demonstrate that LC-R1 outperforms other RL-based and SFT-based methods in both compression rate and accuracy, significantly reducing output tokens with minimal accuracy loss. Our findings provide valuable insights for developing more efficient LRMs that balance computational resource usage with reasoning quality.

1 Introduction

Large Reasoning models (LRMs) have made significant breakthroughs in complex reasoning tasks, which greatly enhances the depth of problem solving by guiding models to generate step-by-step thinking paths (Wei et al., 2023). Recently, OpenAI's O1 (Jaech et al., 2024) have introduced long-thought reasoning models that mimic human-like problem-solving processes. In addition to O1, researchers have also developed models that inference with a similar long-thought reasoning pattern, such as Deepseek-R1 (DeepSeek-AI et al., 2025),

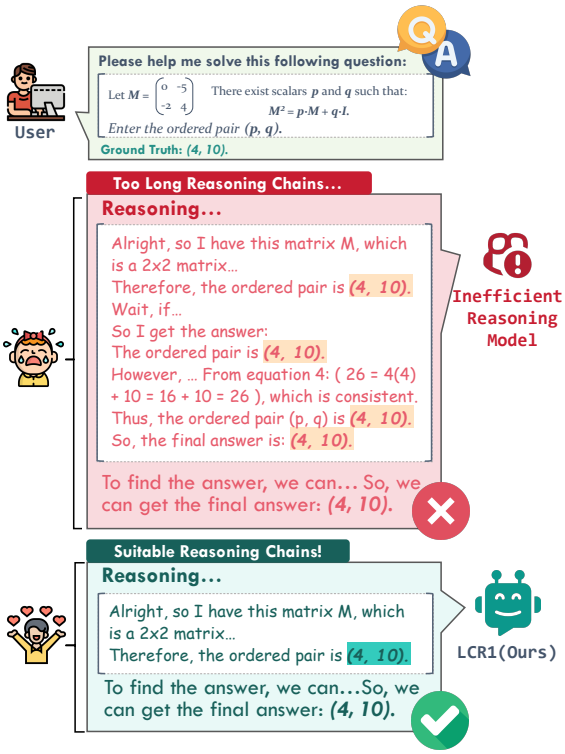

Figure 1: Comparing to other baselines, LC-R1 produces clear explicit responses with less redundant and minimal necessary reasoning paths.

QwQ-32B (Team, 2025b) and Phi-4-Reasoning (Abdin et al., 2025). Trained with Group Relative Policy Optimization (GRPO) using simple rule-based reward, these models demonstrate unprecedented potential by iteratively identifying and correcting errors, simplifying intricate steps, and exploring alternative strategies when initial approaches prove inadequate in fields such as mathematics (Sun et al., 2025) and programming (Gu et al., 2024), marking an important step forward in super-human planning and reasoning skills.

However, with the improvement of *"deep thinking"* ability, an increasingly prominent problem is the consumption of computing resources during the reasoning process (Chen et al., 2025; Aggarwal and Welleck, 2025; Chen et al., 2025). Specifi-

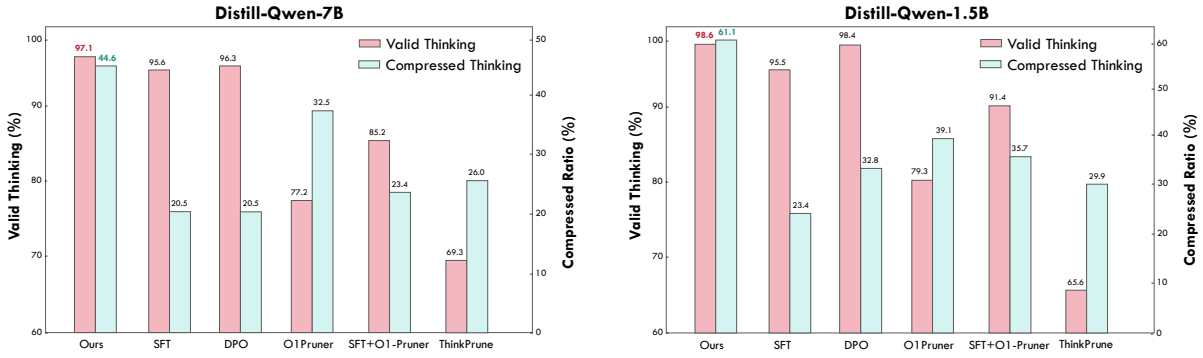

Figure 2: Comparison of different efficient reasoning methods. Our LC-R1 makes the best token compression for current Large Reasoning Models comparing to other Supervised and RL-based Fine-Tuning methods.

cally, existing models tend to generate lengthy and even unnecessary chains of reasoning when solving problems with low complexity or clear solution paths. This phenomenon, referred to by researchers as *"overthinking"*, is manifested in the process of the model consuming far more computing resources than the problem itself requires in reaching the correct conclusion (Chen et al., 2024a; Sui et al., 2025; Cuadron et al., 2025). Therefore, one critical problem arises:

What is the ideal efficient reasoning model?

To address this challenge, we need to establish what constitutes an optimal reasoning/accuracy budget. Therefore, based on model performance and efficiency considerations, we propose three key principles for efficient reasoning:

- **Simplicity**: The proportion of redundant content in thinking process should be minimal, and the model's total reasoning should be concise.
- **Sufficiency**: Model must engage in accurate thinking rather than skipping reasoning steps.
- **Accuracy**: Model must arrive at correct answers as the primary principle.

Based on these three principles, we define two metrics—**Valid Thinking (VT)**—for quantifying performance of efficient reasoning that favor responses exit thinking process after its first outputs the correct answer and overally the complete answer length. And—**Compressed Ratio (CR)**—that measures the efficiency of current length compression methods.

We evaluate current reasoning models and various efficient pruning methods using this metric and discover they fall significantly short of our defined optimal compression ratio, indicating substantial room for improvement. Consequently, guided by our three principles, we design LC-R1, an algorithm based on GRPO design specifically for LRM post-training to enhance reasoning efficiency. We adjust GRPO's loss function, which steers the model to the concise reasoning process. We combine the compressed reward and length reward with GRPO's base reward, guiding the model to pruning the reasoning process from compressing verbose tokens and the rollout length.

We conduct experiments across five challenge mathematical reasoning benchmarks and Distill-Qwen-1.5/7B. Our LC-R1 outperform other RL-based and SFT-based models in compression rate with slight accuracy degradation. Specifically, with only an 4.31% reduction in accuracy, we achieve a 52.83% decrease in length, representing a % improvement over previous *state-of-the-art* methods. We believe our approach can provide methodological and experimental design insights for future RL-based efficient reasoning models.

2 Preliminary: Compression and Efficienct Reasoning Models

2.1 Motivation: Reduce Verbose Thinking

Typical reasoning models operate in a two-phase approach: first "<think>" then perform inference. During the thinking phase, models engage in extensive deliberation to reach an answer, followed by rapid reasoning during the inference phase. This thorough thinking process enables models to correctly solve more challenging problems, achieving higher accuracy rates. However, we've observed that models often derive the correct final answer quite early in their thinking process, yet continue with multiple verification checks to ensure correctness. These verification steps frequently constitute a significant portion of the entire thinking process,

Table 1: Valid Thinking Rate of current *state-of-the-art* Large Reasoning Models. Even the latest Qwen3-32B suffers from a verbosity thinking process.

Model	Avg.	AIME25	AMC	GSM8K	MATH500	OlympiadBench
Qwen-3-32B	57.5	73.8	58.8	53.8	46.6	51.5
QwQ-32B	59.2	70.8	58.2	54.1	53.1	59.6
DeepSeek-R1	65.3	66.5	71.8	64.2	59.8	64.0
Nemotron-Super-49B	60.8	62.1	64.1	63.1	56.6	58.1

resulting in unnecessary verbosity.

Given this phenomenon, we propose a new metric: **Valid Thinking**, defined as the portion of reasoning from the beginning of a model's thinking process until it first derives the correct answer. This definition applies exclusively to CoT (Wei et al., 2023) reasoning that yields correct answers.

LC-EXTRATOR. We develop a specialized model LC-EXTRATOR based on Qwen2.5-3B-Instruct to efficiently extract the position of the first correct answer within the thinking process while maintaining low computational requirements. We construct a dataset consisting of 5,000 <*Question, Thinking Process, Answer*> triplets and identify the position of the first correct token using Gemini-2.5-Flash (Google, 2025a), followed by rigorous rule-based filtering. We then distill this knowledge into a smaller model through training for 2 epochs with these curated samples. LC-EXTRATOR's effectiveness is validated on a 100-sample test set, achieving 98% accuracy as confirmed by human evaluation.

Based on LC-EXTRATOR, we evaluated four state-of-the-art LRMs—QwQ-32b (Team, 2025b), Qwen3-32b (Team, 2025a), Deepseek-R1 (DeepSeek-AI et al., 2025), and Llama-3.3-nemotron-super-49b-v1 (Bercovich et al., 2025)—across AIME25, MATH500, GSM8K, AMC, and OlympiadBench (Sun et al., 2025) benchmarks. Experiment results are under a three time averaged results for robustness.

Table 1 demonstrates that current LRMs (Language Reasoning Models) indeed suffer from severe thinking redundancy issues, presenting significant compression potential. While DeepSeek-R1 outperforms other reasoning models with an average efficiency of 65.3%, there remains substantial room for improvement. Figure 1 reveals that current inefficient reasoning models typically arrive at correct answers during early stages of their thinking process, yet subsequently engage in excessive verification steps and self-doubt that significantly diminish computational efficiency.

2.2 Principles for Efficient Reasoning Model

By examining prior work and efficiency/accuracy tradeoffs, we establish key guidelines for truly efficient reasoning models:

- **Simplicity:** Minimal redundancy in thinking processes with concise total reasoning length. This addresses computational inefficiency of *"overthinking,"* where models generate excessive explanations. We quantify this through compression metrics measuring essential-to-total reasoning ratios.
- **Sufficiency:** Accurate thinking without skipping critical reasoning steps. Brevity must not compromise logical completeness. We evaluate by tracking whether key logical steps remain intact after compression.
- **Accuracy:** Correct answers as the primary constraint—efficiency gains must not compromise solution correctness. Measured through standard accuracy metrics across reasoning benchmarks.

These principles require models to maintain critical reasoning paths while eliminating redundant verifications and circular thinking.

3 LC-R1: Length Compression with Efficient Reasoning Principles

In this section, we introduce our LC-R1 method whose pipeline is shown in 3.

3.1 Problem Formulation

Let $\mathcal{M}$ be the model and q be the given query. The output is $o \sim \mathcal{M}(q)$, where $o = \text{cat}(R, A)$ consists of a reasoning part R and an answer part A. The function $t(o) = R$ extracts the reasoning part. For a reasoning part R, its effective prefix R' includes the content from the beginning of R up to the first occurrence of the correct answer. If R does not contain the correct answer, then $R' = R$. The function $f(\{R, A\}) = \{R', A\}$ extracts the

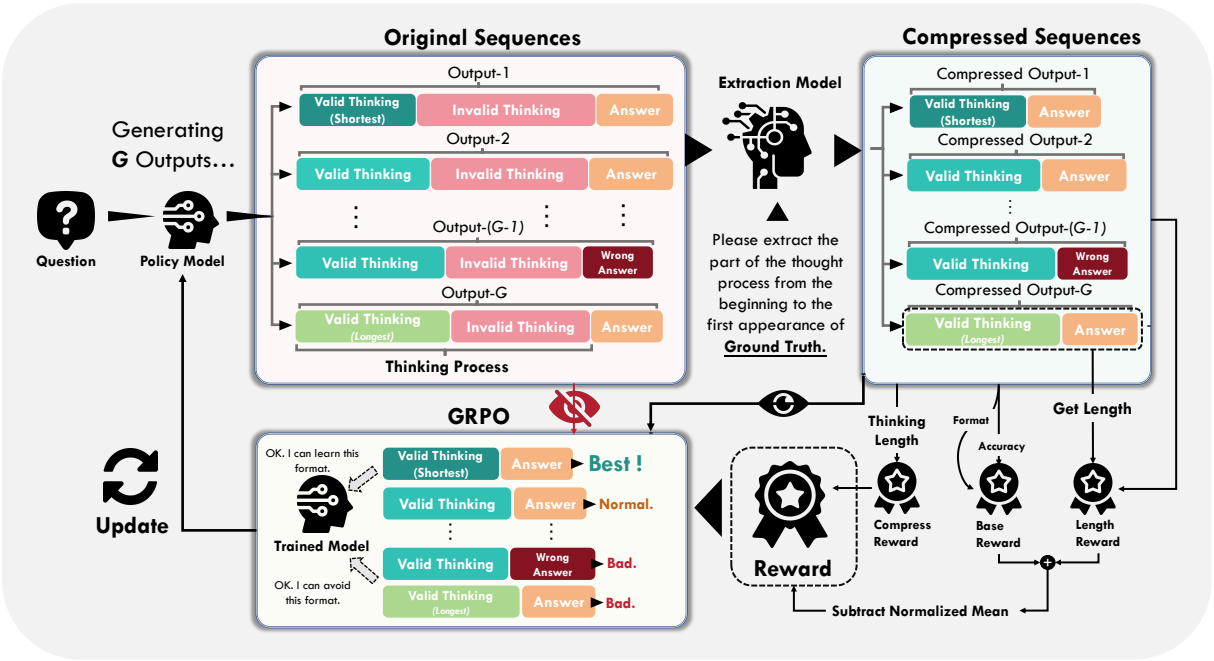

Figure 3: **An overview of our proposed LC-R1 method.** LC-R1 consists of two key steps: ***(1)* extraction**. An extraction model extracts the valid Thinking to generate compressed sequences. ***(2)* Getting reward.** Compressed sequences are used to calculate Length reward and compress reward, getting the Advantages of sequences. ***(3)* LC-GRPO**. GRPO loss is calculated by compressed sequences, steering models to get concise reasoning process.

concise reasoning part and concatenates it with the answer. We denote o_i as an original model output and $o_i' = f(o_i)$ as the refined, compressed output.

LC-R1 is a method based on GRPO to compress the reasoning process efficiently. Within a group, let $\mathcal{C}$ denote the set of indices for sequences o_i that are considered *"correct"* (*e.g.*, leading to a correct final answer and exhibiting sound reasoning), and $\mathcal{W}$ be the set of indices for "wrong" or incorrect sequences.The total number of sequences in a batch is $G = |\mathcal{C}| + |\mathcal{W}|$.

3.2 Reward and Objective Design

Our method can primarily be divided into two aspects: the Length Reward, aimed at reducing the overall output length, and the Compress Reward, aimed at compressing redundant parts of the model's reasoning.

Length Reward. To compress the overall length of the model output, we propose adding a length penalty during the GRPO training process. We hope that the correct sequences in a group are as short as possible. For a given problem, we set a threshold based on the problem's difficulty. We denote a bool value $b = \text{mean}_{j\in\mathcal{C}}|o_j| > \text{threshold}$, and we have:

$$r_{i,\text{length}} = \begin{cases} 1 - \frac{|o_i'|}{\max_{j\in\mathcal{C}}|o_j'|}, & \text{if } i \in \mathcal{C} \,\&\, b \\ 0, & \text{if } i \in \mathcal{W} \end{cases} \quad (1)$$

In the formula, we utilize the maximum length within a group to adaptively adjust the length coefficient. Unlike Kimi (Team et al., 2025), we do not use min-max normalization, thus avoiding the amplification of subtle differences in length, which ensures the focus remains on problems with significant length disparities within a group. Additionally, if the mean length of sequences in a group is less than the threshold, no Length Reward is given to prevent excessive compression by the model. Next, based on the Length Reward and the original base reward, we can obtain the combined reward:

$$r_{i,\text{base}} = r_{i,\text{format}} + r_{i,\text{accuracy}} \quad (2)$$

$$\tilde{r}_i = \begin{cases} r_{i,\text{base}} + \alpha \cdot r_{i,\text{length}}, & \text{if } i \in \mathcal{C} \\ r_{i,\text{base}}, & \text{if } i \in \mathcal{W} \end{cases} \quad (3)$$

$$r = \tilde{r}_i - \text{mean}(\{\tilde{r}_j\}_{j=1}^{G}) \quad (4)$$

We only perform mean-subtraction normalization on the combined reward, also to prevent the model from being biased by difficulty due to standardization when the length differences are too small.

Compress Reward. For the original GRPO method, the loss calculation is based on the model's own sampling results. To compress redundant tokens in the model's reasoning stage and learn to stop reasoning upon first reaching the ground truth, we modify the GRPO formula as follows:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim P(Q),\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(O|q)} \quad (5)$$
$$\Bigg[\frac{1}{\sum_{i=1}^{G}|o_i'|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i'|}\Bigg\{\min\Big[R_t(\theta)\cdot\hat{A}_i,$$
$$\text{clip}\Big(R_t(\theta),\,1-\epsilon,\,1+\epsilon\Big)\cdot\hat{A}_i\Big]$$
$$-\beta\,D_{\text{KL}}\big(\pi_\theta(\cdot|q)\,\|\,\pi_{\text{ref}}(\cdot|q)\big)\Bigg\}\Bigg]$$

Where $o_i' = f(o_i)$, that is we use the compressed sequences to calculate loss, and we use another model to fit the function. and $R_t(\theta)$ is defined as:

$$R_t(\theta) = \frac{\pi_\theta(o_{i,t}'|q, o_{i,<t}')}{\pi_{\theta_{\text{old}}}(o_{i,t}'|q, o_{i,<t}')} \quad (6)$$

We define the Advantages as follow:

$$\hat{A}_i = (\tilde{r}_i - \text{mean}(\{\tilde{r}_j\}_{j=1}^{G})) + r_{i,\text{compress}} \quad (7)$$

$$r_{i,\text{compress}} = \begin{cases} 1 - \frac{|t(o_i')|}{|t(o_i)|}, & \text{if } i \in \mathcal{C} \;\&\; \text{ans} \in t(o_i') \\ -1, & \text{if } i \in \mathcal{C} \;\&\; \text{ans} \notin t(o_i') \\ 0, & \text{if } i \in \mathcal{W} \end{cases} \quad (8)$$

In the Advantages, we add an additional reward $r_{i,\text{compress}}$ on top of the original normalized reward. The reason for this design is that the current model's loss calculation is based on the compressed sequence o_i'. To enable the model to learn strategies for compressing the reasoning part, o_i' needs to have a generally positive advantage on early . We utilize $1 - \frac{|t(o_i')|}{|t(o_i)|}$ to steer the model towards more compressed sequences.

Based on the principle of Sufficiency, the model should engage in sufficient reasoning during the reasoning stage. Therefore, for cases where the correct answer is not obtained during the reasoning stage, we consider the reasoning to be insufficient and impose a larger penalty, which lies a robustness for training process.

What's more,we drew inspiration from the work of DAPO (Yu et al., 2025), modifying GRPO to calculate the mean token reward across all tokens in a group, instead of averaging the token rewards within a single sequence. which eliminates the original GRPO method's preference for short sequences, facilitating the validation of our method's effectiveness.

4 Experiments

4.1 Experiment Setups

Backbone Models. We choose DeepSeek-R1-Distill-Qwen-7B and DeepSeek-R1-Distill-Qwen-1.5B to be the backbone models.

Dataset. We used a mixed-difficulty dataset, combining past AIME competition problems with the MATH dataset in a 1:3 ratio to create 1500 training samples. This approach enables the model to learn length compression across problems of varying difficulty.

Evaluation. We test our model's performance on multiple datasets, including AIME25, MATH500, GSM8K, AMC, and OlympiadBench. We use averaged Pass@1 as our primary metric. For each test, we sample N times, setting top-p=0.95 and temperature=0.7. For AIME25, we set $N = 16$, while for the other test sets, we set $N = 8$. We set the maximum length to 16384. Additionally, we calculate their mean as a comprehensive evaluation of the model.

4.2 Baselines

SFT. OVERTHINK (Chen et al., 2024a) proposes using the first solution for SFT to significantly reduce model length. We reconstructe an SFT training set from the previously constructed label dataset, with the think portion containing only label data, using a total of 5000 samples for training.

DPO (Rafailov et al., 2023). We sample the model multiple times on 5000 MATH benchmark problems, taking the shortest and longest samples as positive and negative samples, respectively, and use 5000 samples for training.

O1 Pruner (Luo et al., 2025b). This work employed a PPO-like offline fine-tuning method to significantly compress chain-of-thought (CoT) length across multiple benchmarks while maintaining performance. We similarly use 5000 samples from the MATH dataset to train the model.

THINKPRUNE (Hou et al., 2025). This work utilized a reinforcement learning approach, designing a length-clip reward to compress CoT length in multiple stages. We use the open-source Length3000

Table 2: Accuracy (above) and length (below) of models and methods on different benchmarks. Avg represents change compared to the large reasoning model (+ increase, – decrease).

	Distill-Qwen-7B						Distill-Qwen-1.5B					
Method	**AIME25**	**MATH500**	**GSM8K**	**Olympiad**	**AMC**	**Avg. (%)**	**AIME25**	**MATH500**	**GSM8K**	**Olympiad**	**AMC**	**Avg. (%)**
Origin	40.2 (11005)	93.0 (3880)	92.6 (1787)	61.2 (7388)	81.9 (6689)	–	22.8 (12129)	83.7 (4869)	83.4 (2294)	44.2 (9258)	61.2 (8696)	–
SFT	36.6 (9457)	90.2 (2497)	91.9 (946)	56.0 (6329)	78.7 (5231)	–4.20% (–20.45%)	20.5 (10639)	81.4 (3045)	81.3 (1134)	42.7 (7637)	59.7 (6608)	–3.28% (–23.42%)
DPO	36.9 (9718)	91.4 (2277)	90.3 (980)	56.2 (6338)	78.6 (5122)	–4.20% (–20.53%)	19.4 (10316)	79.0 (2749)	80.9 (855)	41.1 (6544)	56.7 (5912)	–6.16% (–32.80%)
O1-Pruner	35.0 (8263)	91.5 (2268)	91.1 (1012)	59.6 (4712)	77.1 (4510)	–3.96% (–32.50%)	24.1 (8687)	84.3 (2913)	82.7 (1162)	47.0 (5960)	69.3 (5193)	**+4.10%** (–39.08%)
ThinkPrune	37.6 (8431)	91.9 (2631)	91.4 (1092)	58.9 (5732)	78.1 (4881)	**–2.98%** (–25.96%)	19.4 (8851)	83.1 (3517)	84.6 (1533)	43.0 (6180)	57.6 (6070)	–2.68% (–29.89%)
SFT+O1-Pruner	35.5 (9466)	91.0 (2245)	89.7 (920)	56.0 (5807)	76.6 (5133)	–5.45% (–23.36%)	17.5 (9075)	80.2 (2769)	81.5 (919)	40.0 (6411)	58.7 (5553)	–5.89% (–35.71%)
LC-R1 (Ours)	35.6 (6911)	90.6 (1843)	90.9 (675)	57.8 (4378)	78.8 (3799)	–4.12% **(–44.56%)**	20.8 (5953)	79.3 (1822)	80.2 (621)	42.7 (3780)	59.0 (3591)	–4.50% **(–61.10%)**

dataset from this work, test THINKPRUNE-3k, and set parameters group=8 and epoch=2.0.

SFT + O1-Pruner (Luo et al., 2025b). To better demonstrate the effectiveness of our method, we also compare it with a strong two-stage training approach combining SFT and O1-Pruner.

4.3 Experiment Results

LC-R1 outperform other baselines a large margin with less tokens and comparative performance. From Table 2, our method achieve better results on both two models. Based on the test results, most fine-tuning methods had a similar impact on the model's accuracy across various benchmarks. Among these methods, LC-R1 achieved the greatest length reduction, compressing the reasoning length by 44.56% and 61.10% on 7B and 1.5B, respectively. Additionally, compared to the SFT+O1-Pruner method, it is evident that using existing methods to first compress redundant tokens and then applying RL methods to shorten CoT length does not effectively reduce the CoT length of the reasoning model.

Combining length and compress reward brings superior efficiency reasoning. Our ablation study primarily focused on the Length Reward and Compress Reward. To understand the individual contributions of these two components to our proposed method, we conduct ablation studies on both models.

As shown in Table 1, training with either component alone achieved good compression results. For instance, on DeepSeek-R1-Distill-Qwen-7B, the effects of both components were comparable to our overall baseline performance, while on DeepSeek-R1-Distill-Qwen-1.5B, both achieve better results than the baseline. However, combining both components for training resulted in a greater compression ratio with only a slight reduction in accuracy. Therefore, both modules are relatively important to our method.

5 Discussion and Analysis of Compression

5.1 Compression Ratio

To investigate whether our method effectively compresses the redundant parts of the reasoning process, we tested the results of different methods trained on two models, as shown in Figure 2.

The results clearly demonstrate that our method achieve excellent performance in compressing redundant parts of the reasoning process, with a high compression ratio for the overall chain-of-thought (CoT) compared to the original model. The SFT method also achieved a high compression ratio for redundant reasoning parts, but its overall CoT compression ratio was lower, because it is unable for the sft model to produce outputs shorter than training dataset. Other non-SFT methods, such as O1-Pruner and ThinkPrune, showed lower compression ratios for redundant reasoning, indicating that these methods still have room for further compression.

We count tokens associated with long CoT, with our method outperforming others, as shown in Figure 6. The token list is in Table 4.

5.2 Impact of Compression on Performance

To investigate the compressing impact of test-time scaling capability of reasoning models, we evaluate on Pass@k metric on AIME25 benchmark for

Table 3: Accuracy (above) and length (below) of models and methods on different benchmarks. Avg represents change compared to the large reasoning model (+ increase, – decrease).

Method	Distill-Qwen-7B						Distill-Qwen-1.5B					
	AIME25	MATH500	GSM8K	Olympiad	AMC	Avg. (%)	AIME25	MATH500	GSM8K	Olympiad	AMC	Avg. (%)
LC-R1(Ours)	35.6 (6911)	90.6 (1843)	90.9 (675)	57.8 (4378)	78.8 (3799)	–4.12% (**–44.56%**)	20.8 (5953)	79.3 (1822)	80.2 (621)	42.7 (3780)	59.0 (3591)	–4.50% (**–61.10%**)
wo L-reward	39.1 (9625)	91.3 (2316)	90.6 (696)	59.4 (5779)	79.0 (5021)	–2.58% (–23.79%)	21.3 (7061)	81.2 (2270)	83.3 (754)	43.4 (5024)	63.1 (4433)	–1.02% (–50.21%)
wo C-reward	38.3 (8474)	92.9 (2498)	91.1 (1012)	59.1 (5344)	80.5 (4741)	**–1.90%** (–28.24%)	21.9 (7988)	83.2 (2965)	84.1 (1160)	44.0 (5608)	66.1 (5192)	**+1.35%** (–41.62%)

Figure 4: A case study comparing LC-R1 (Ours) with O1-Pruner. We advice a **ZOOM-IN** for a closer look. When answering the same question, LC-R1 achieves 100% valid ratio with 1324 tokens consumption (875 tokens for valid thinking, 449 tokens for final response) while O1-Pruner consumes 2119 tokens (800 tokens for valid thinking, 902 tokens for invalid thinking and 417 tokens for final response).

models before and after CoT compression. We select three models based on CoT length—short, medium, and long—namely LC-R1, SFT, and Origin. We sample the models 128 times and calculate the pass@k results for k ranging from 1 to 128.

Figure 5 shows that compressing the CoT length does not affect the model's potential. This further indicates that our method has minimal impact on the model's performance and also confirms that the redundant reasoning parts compressed by our method indeed have trivial contributions to the model's ability to produce correct answers.

6 Related Work

Large Reasoning Model. Research on advanced reasoning in LLMs (Team, 2024a) has focused on scaling computation (Chen et al., 2024b; Snell et al., 2024) and refining inference generation. Techniques range from Chain-of-Thought (CoT) prompting (Wei et al., 2023) to Process Re-

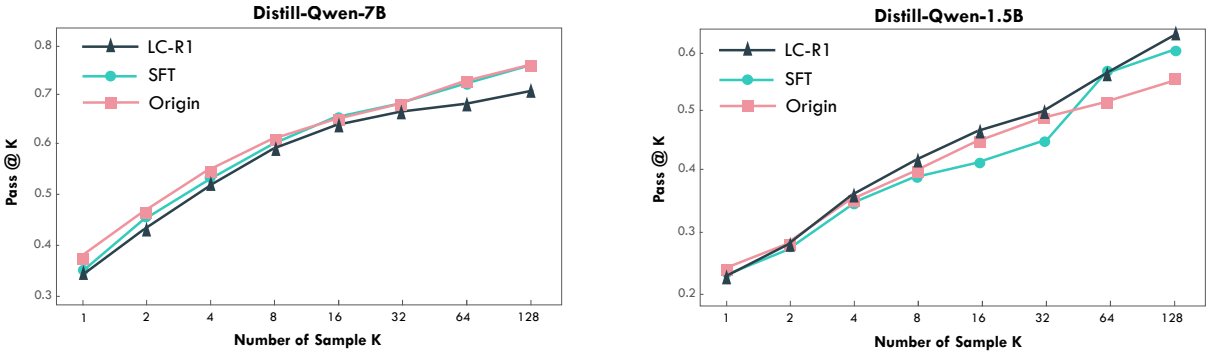

Figure 5: Comparison of different efficient reasoning methods. Our LC-R1 make the best token compression for current Large Reasoning Models comparing to other Supervised and RL-based Fine-Tuning methods.

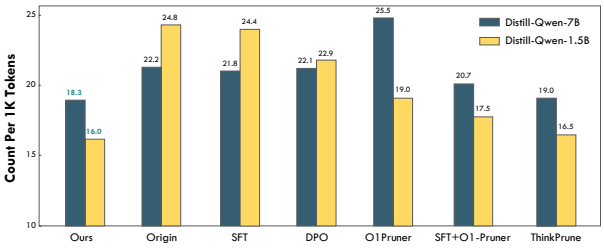

Figure 6: Across all benchmark tests for Distill-Qwen-7B/1.5B, LC-R1 uses the fewest tokens per thousand—meaning it produces the least invalid reasoning.

ward Models and search-guided decoding (Brown et al., 2024) for aggregating reasoning paths. These advances produced powerful Large Reasoning Models (LRMs) like ChatGPT-O1 (OpenAI, 2024), Deepseek-R1 (DeepSeek-AI et al., 2025), QwQ (Team, 2025b), and Gemini2.5 (Google, 2025b), which spontaneously generate extensive CoT with thinking, backtracking, and verification. Open-source models derive reasoning abilities through reinforcement learning (RL) (DeepSeek-AI et al., 2025; Ramesh et al., 2024; Muennighoff et al., 2025) or distillation (DeepSeek-AI et al., 2025; Yu et al., 2024) from RL-produced CoT data, with recent work (Yue et al., 2025) analyzing differences between these approaches.

Efficient Reasoning. While elaborate reasoning enhances performance, its verbosity creates efficiency challenges (Chen et al., 2024a), increasing inference latency and computational costs. Research on efficient reasoning seeks to reduce reasoning trace length without sacrificing accuracy. Approaches include CoT optimization (Aggarwal and Welleck, 2025; Luo et al., 2025b; Shen et al., 2025) through RL with length-based rewards (Sun et al., 2024; Liao et al., 2025; Luo et al., 2025b; Aggarwal and Welleck, 2025; Luo et al., 2025a) and fine-tuning with variable-length CoT data (Han et al., 2024; Yu et al., 2024; Munkhbat et al., 2025). Training-free strategies employ dynamic reasoning during inference (Yang et al., 2025a; Zhang et al., 2025; Wu et al., 2025; Lin et al., 2025) or prompt-guided efficient reasoning (Cheng and Van Durme, 2024; Xu et al., 2025; Han et al., 2024; Ma et al., 2025).

Overthinking. Recent studies examine generated thought processes, particularly *Aha Moments* (DeepSeek-AI et al., 2025; Liu et al., 2025) marked by keywords like *"wait"* and *"hmm"*, which indicate self-reflection (Chen et al., 2025) allowing models to reassess reasoning paths. Research (Yang et al., 2025b; Zhang et al., 2025) has begun characterizing these moments and exploring mechanisms behind such spontaneous self-reflection. However, frequent occurrences of these keywords can lead to *Overthinking* (Chen et al., 2024a; Sui et al., 2025), where models continue reflecting after reaching correct conclusions.

7 Conclusion

We introduce LC-R1, an algorithm designed to address the efficient reasoning problem by optimizing length compression while maintaining reasoning accuracy. We establish three fundamental principles for efficient reasoning—Simplicity, Sufficiency, and Accuracy—and proposed two metrics, Valid Thinking and Compressed Ratio, to quantitatively evaluate reasoning efficiency. Our experimental results across five mathematical reasoning benchmarks demonstrate that LC-R1 significantly outperforms existing pruning-based and SFT-based methods, providing valuable insights for developing more resource-efficient AI systems.

Limitation

Our current experimental scope focused on 1.5B and 7B models due to computational considerations, with larger model scales representing promising avenues for future investigation. Additionally, while our reward function design incorporates several hyperparameters—particularly the balancing factors between length constraint rewards—we maintained consistent settings across experiments due to computational efficiency considerations. In future work, we plan to further explore the optimal balance between reasoning trace length and accuracy, as well as investigate enhanced reward formulations that could potentially yield more efficient reasoning capabilities.

References

Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, Piero Kauffmann, Yash Lara, Caio César Teodoro Mendes, Arindam Mitra, Besmira Nushi, Dimitris Papailiopoulos, Olli Saarikivi, Shital Shah, Vaishnavi Shrivastava, and 4 others. 2025. Phi-4-reasoning technical report. *Preprint*, arXiv:2504.21318.

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

Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, Ido Shahaf, Oren Tropp, Ehud Karpas, Ran Zilberstein, Jiaqi Zeng, Soumye Singhal, Alexander Bukharin, ... Yian Zhang, and Chris Alexiuk. 2025. Llama-nemotron: Efficient reasoning models. *Preprint*, arXiv:2505.00949.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *Preprint*, arXiv:2407.21787.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *Preprint*, arXiv:2503.09567.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024a. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, and 1 others. 2024b. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Jeffrey Cheng and Benjamin Van Durme. 2024. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*.

Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, and 1 others. 2025. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao ..., and Zhen Zhang. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Google. 2025a. Gemini 2.5 flash. https://developers.googleblog.com/en/start-building-with-gemini-25-flash/.

Google. 2025b. Gemini 2.5 pro. https://cloud.google.com/vertex-ai/generative-ai/docs/models/gemini/2-5-pro.

Alex Gu, Baptiste Rozière, Hugh Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida I. Wang. 2024. Cruxeval: A benchmark for code reasoning, understanding and execution. *arXiv preprint arXiv:2401.03065*.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2024. Token-budget-aware llm reasoning. *arXiv preprint arXiv:2412.18547*.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. 2025. Thinkprune: Pruning long chain-of-thought of llms via reinforcement learning. *arXiv preprint arXiv:2504.01296*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Baohao Liao, Yuhui Xu, Hanze Dong, Junnan Li, Christof Monz, Silvio Savarese, Doyen Sahoo, and Caiming Xiong. 2025. Reward-guided speculative decoding for efficient llm reasoning. *Preprint*, arXiv:2501.19324.

Kevin Lin, Charlie Snell, Yu Wang, Charles Packer, Sarah Wooders, Ion Stoica, and Joseph E. Gonzalez. 2025. Sleep-time compute: Beyond inference scaling at test-time. *Preprint*, arXiv:2504.13171.

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

Haotian Luo, Haiying He, Yibo Wang, Jinluan Yang, Rui Liu, Naiqiang Tan, Xiaochun Cao, Dacheng Tao, and Li Shen. 2025a. Adar1: From long-cot to hybrid-cot via bi-level adaptive reasoning optimization. *arXiv preprint arXiv:2504.21659*.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025b. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *Preprint*, arXiv:2501.12570.

Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei Zaharia. 2025. Reasoning models can be effective without thinking. *Preprint*, arXiv:2504.09858.

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. *Preprint*, arXiv:2501.19393.

Tergel Munkhbat, Namgyu Ho, Seo Hyun Kim, Yongjin Yang, Yujin Kim, and Se-Young Yun. 2025. Self-training elicits concise reasoning in large language models. *Preprint*, arXiv:2502.20122.

OpenAI. 2024. Chatgpt. `https://openai.com/o1/`.

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. *Advances in Neural Information Processing Systems*, 36:53728–53741.

Shyam Sundhar Ramesh, Yifan Hu, Iason Chaimalas, Viraj Mehta, Pier Giuseppe Sessa, Haitham Bou Ammar, and Ilija Bogunovic. 2024. Group robust preference optimization in reward-free rlhf. *Advances in Neural Information Processing Systems*, 37:37100–37137.

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, and Shiguo Lian. 2025. Dast: Difficulty-adaptive slow-thinking for large reasoning models. *Preprint*, arXiv:2503.04472.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Hanjie Chen, Xia Hu, and 1 others. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*.

Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. 2024. Fast best-of-n decoding via speculative rejection. *Preprint*, arXiv:2410.20290.

Haoxiang Sun, Yingqian Min, Zhipeng Chen, Wayne Xin Zhao, Zheng Liu, Zhongyuan Wang, Lei Fang, and Ji-Rong Wen. 2025. Challenging the boundaries of reasoning: An olympiad-level math benchmark for large language models. *Preprint*, arXiv:2503.21380.

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, Congcong Wang, Dehao Zhang, Dikang Du, Dongliang Wang, Enming Yuan, Enzhe Lu, Fang Li, Flood Sung, Guangda Wei, and 73 others. 2025. Kimi-VL technical report. *Preprint*, arXiv:2504.07491.

OpenAI Team. 2024a. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Qwen Team. 2024b. Qwen2.5: A party of foundation models.

Qwen Team. 2025a. Qwen3.

Qwen Team. 2025b. Qwq-32b: Embracing the power of reinforcement learning.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Yuyang Wu, Yifei Wang, Tianqi Du, Stefanie Jegelka, and Yisen Wang. 2025. When more is less: Understanding chain-of-thought length in llms. *Preprint*, arXiv:2502.07266.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. 2025. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Zheng Lin, Li Cao, and Weiping Wang. 2025a. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*.

Shu Yang, Junchao Wu, Xin Chen, Yunze Xiao, Xinyi Yang, Derek F. Wong, and Di Wang. 2025b. Understanding aha moments: from external observations to internal mechanisms. *Preprint*, arXiv:2504.02956.

Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024. Distilling system 2 into system 1. *Preprint*, arXiv:2407.06023.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

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

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. 2025. Reasoning models know when they're right: Probing hidden states for self-verification. *Preprint*, arXiv:2504.05419.

A Details of LC-Extractor

We train Qwen-2.5-3B-Instruct (Team, 2024b) as the LC-Extractor model. Our LC-Extractor model is activate by the prompt in Figure 7. We also design the annotation tool in Figure 8 to evaluate the model. It achieves 98.0% accuracy.

B Detailed Experiment Setups

B.1 Model

We use **DeepSeek-R1**(DeepSeek-AI et al., 2025), **Qwen3-32B**(Team, 2025a), **QwQ-32B**(Team, 2025b), **Llama-3.3-Nemotrom-Super-49B-V1**(Bercovich et al., 2025), **Distill-Qwen-7B**, **Distill-Qwen-1.5B**(Yu et al., 2024), and **Qwen-2.5-3B-Instruct**(Team, 2024b) models in our paper. We introduce their licenses and key characteristics as follows:

- **DeepSeek-R1.** An open-source 671 B→37 B MoE reasoning model trained largely through reinforcement learning, which elicits self-verification, reflection and lengthy chain-of-thought traces while supporting 128K-token context; it matches proprietary o1 on math / code benchmarks using only public data.
- **Qwen3-32B.** The 32.8 B-parameter third-generation Qwen model that toggles between "thinking" and "non-thinking" modes, delivering state-of-the-art reasoning, multilingual chat and up to 131 K context in a single dense checkpoint.
- **QwQ-32B.** A medium-sized Qwen reasoning variant refined with SFT + RL; provides explicit <think> traces, 131 K context and DeepSeek-R1–level accuracy on hard evaluations.
- **Llama-3.3-Nemotrom-Super-49B-V1.** NVIDIA's NAS-pruned 49 B derivative of Llama-3.3-70B, post-trained for reasoning, RAG and tool calling; couples 128 K context with single-H100 deployment efficiency for cost-sensitive production.
- **Distill-Qwen-7B.** A 7 B dense checkpoint distilled from DeepSeek-R1 onto the Qwen2.5 backbone, pushing small-model MATH-500 pass1 beyond 92 % and surpassing o1-mini on several reasoning suites while remaining laptop-friendly.
- **Distill-Qwen-1.5B.** An ultra-compact 1.5 B model distilled from R1 that preserves chain-of-thought and achieves 83.9 % pass1 on MATH-500, bringing competitive analytical power to edge and mobile deployments.
- **Qwen-2.5-3B-Instruct.** A 3.09 B instruction-tuned model with 128 K context, strengthened coding/math skills and multilingual support, designed as a lightweight yet controllable chat foundation for downstream tasks.

B.2 Dataset

We benchmark on the **AIME25**, **MATH500**, **GSM8K**, **Olympiad** (Sun et al., 2025), and **AMC** benchmarks in our paper. We introduce them as follows:

- **AIME25.** A benchmark with 30 questions distilled from twenty-five years of *American Invitational Mathematics Examination* papers. Each item is a three-digit short-answer problem that probes upper-secondary algebra, geometry, combinatorics.
- **MATH500.** A 500-problem evaluation slice covering the full subject breadth of the original *MATH* competition corpus. Balanced across difficulty tiers and topics, it serves as a rigorous yardstick for advanced high-school and early undergraduate mathematical reasoning, without the runtime burden of the complete 12k-question set.
- **GSM8K.** The widely-adopted *Grade-School Math 8K* benchmark of 1,319 everyday word-problems. Requiring multi-step arithmetic and commonsense, GSM8K remains the de-facto standard for assessing chain-of-thought quality on conversational math tasks.
- **Olympiad.** A curated collection of roughly 3 k national and international mathematics-olympiad problems. Predominantly proof-style or numeric-answer challenges, this benchmark gauges creative, non-routine reasoning at the highest pre-university level.

Prompt to Extract Answer Prefix

You are Qwen, created by Alibaba Cloud. You are a helpful assistant.

Instruction:

Extract Answer Prefix You'll get a Problem, a Thinking Process, and its Ground Truth Answer.

Your Task:

1. Read the Thinking Process from the beginning carefully.
2. Find the first sentence that reveals the Ground Truth Answer.
3. Copy everything from the start of the Thinking Process up to and including that sentence.
4. Important: Do not include any text after that sentence.

Example:

- Problem: What is 1 + 1?
- Thinking Process: Okay, I need to solve 1 + 1. That gives 2. Let me check again–yes, it's 2.
- Ground Truth Answer: 2.
- Expected Output: Okay, I need to solve 1 + 1. That gives 2.

Input Provided:

- Problem: <Problem>
- Thinking Process: <Thinking Process>
- Ground Truth Answer: <Ground Truth Answer>

Your Output:

A prefix of "Thinking Process", with Ground Truth at the end.

Figure 7: Our prompt for extraction of answer prefix.

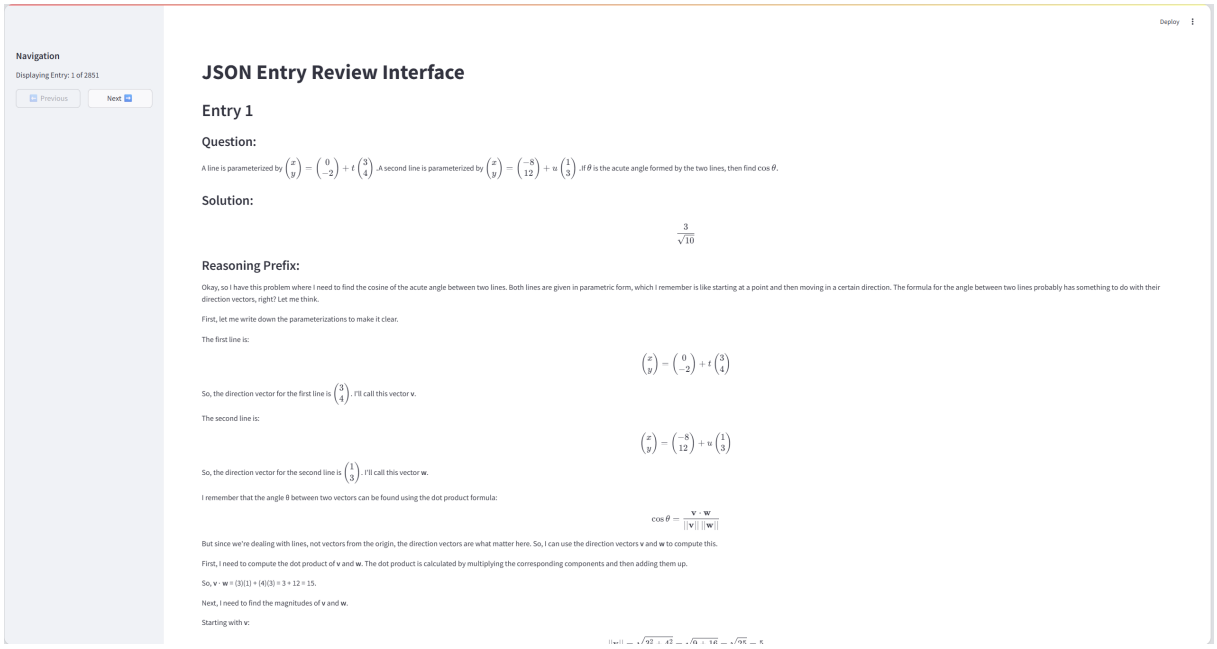

Figure 8: The annotation tool to evaluate the LC-Extratcor.

- **AMC.** An aggregate of 83 from the *American Mathematics Competitions 10/12*. Spanning 2000–2024, it offers a longitudinal benchmark on foundational secondary-school math.

B.3 Reasoning Token list

Table 4: Keyword List for Suppressing.

Keyword List for Suppressing
“wait", “alternatively", “hmm", “but", “however", “alternative", “another", “check", “double-check", “oh", “maybe", “verify", “other", “again", “now", “ah", “any"

C Case Study

We make some case studies to compare LC-R1 with other method. These case studies are shown in Figure 9.

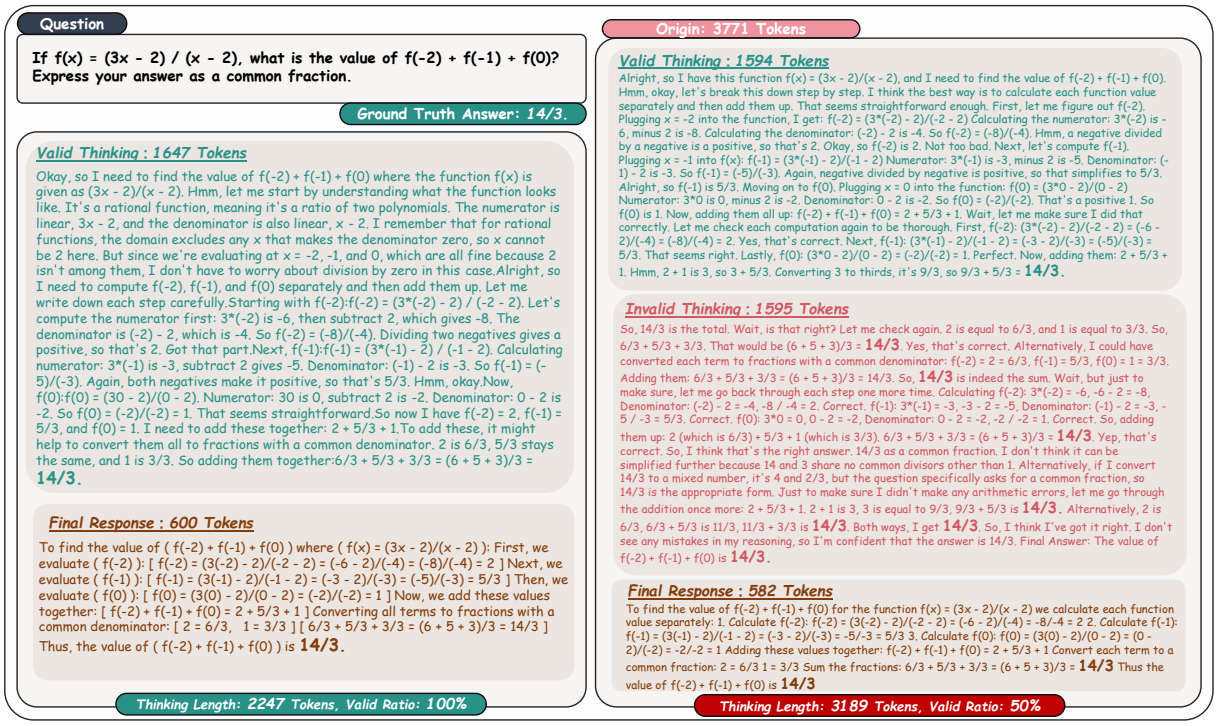

Figure 9: More case studies.