Think Twice: Enhancing LLM Reasoning by Scaling Multi-round Test-time Thinking

Xiaoyu Tian, Sitong Zhao, Haotian Wang, Shuaiting Chen, Yunjie Ji, Yiping Peng, Han Zhao, and Xiangang Li

a-m-team

Abstract

Recent advances in large language models (LLMs), such as OpenAI-o1 and DeepSeek-R1, have demonstrated the effectiveness of test-time scaling, where extended reasoning processes substantially enhance model performance. Despite this, current models are constrained by limitations in handling long texts and reinforcement learning (RL) training efficiency. To address these issues, we propose a simple yet effective test-time scaling approach—**Multi-round Thinking**. This method iteratively refines model reasoning by leveraging previous answers as prompts for subsequent rounds. Extensive experiments across multiple models, including QwQ-32B and DeepSeek-R1, consistently show performance improvements on various benchmarks such asAIME 2024, MATH-500, GPQA-diamond, and LiveCodeBench. For instance, the accuracy of QwQ-32B improved from 80.3% (Round 1) to 82.1% (Round 2) on theAIME 2024 dataset, while DeepSeek-R1 showed a similar increase from 79.7% to 82.0%. These results confirm that **Multi-round Thinking** is a broadly applicable, straightforward approach to achieving stable enhancements in model performance, underscoring its potential for future developments in test-time scaling techniques.

The key prompt:

```
Original question prompt
The assistant's previous answer is: <answer> last round answer </answer>, and please
re-answer.
```
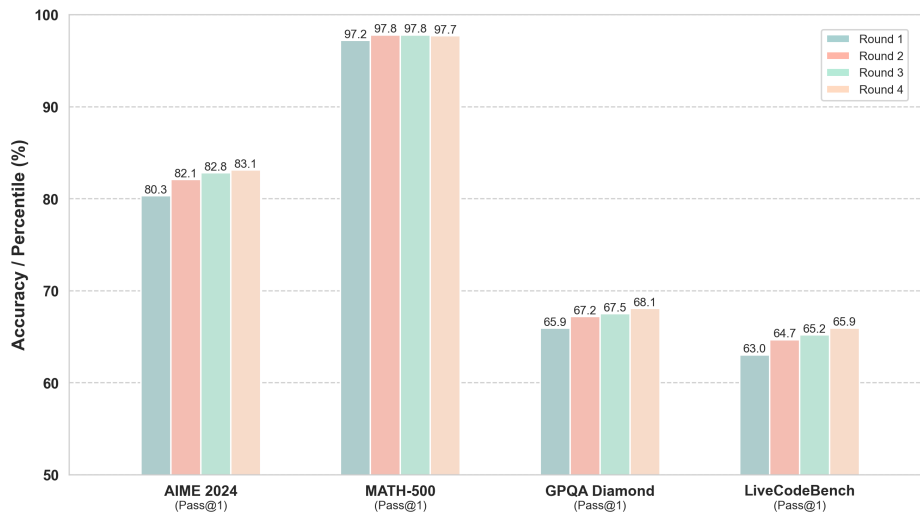

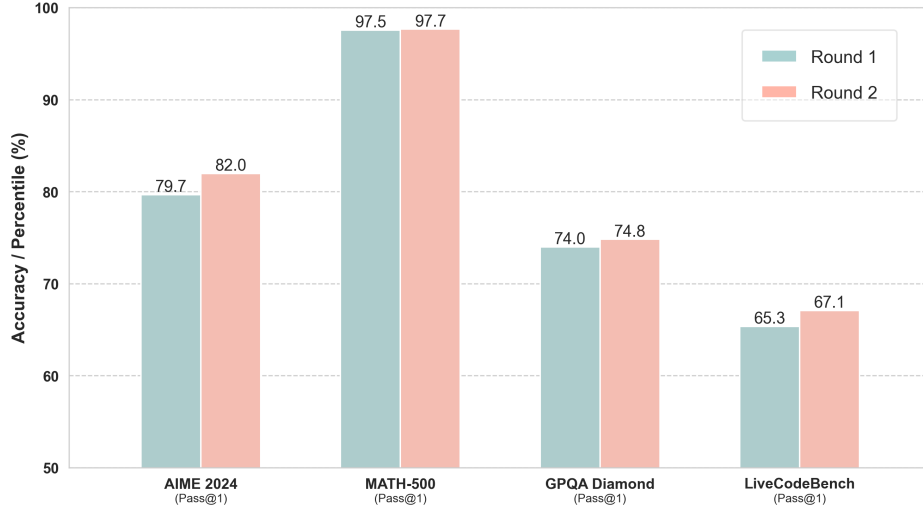Figure 1: Benchmark performance of QwQ-32B using Multi-round Thinking.

Figure 2: Benchmark performance of DeepSeek-R1 using Multi-round Thinking.

1 Introduction

Inference test-time compute (Yang et al., 2025; Wu et al., 2025) refers to the computational resources utilized by large language models (LLMs) during the generation of prompt responses, distinct from the training compute used for model creation and refinement. Leveraging step-by-step reasoning has shown substantial improvements in solving complex tasks by explicitly providing models with intermediate reasoning steps(Lightman et al., 2023; Wei et al., 2023), significantly enhancing accuracy.

In recent years, the performance improvements of language models have largely depended on massive-scale self-supervised pre-training (Kaplan et al., 2020; Hoffmann et al., 2022), scaling up training-time compute. However, as advancements in training-time scaling slow, increasing attention is turning towards scaling up test-time compute (Muennighoff et al., 2025; Chen et al., 2025). OpenAI (OpenAI, 2024) pioneered this approach with their o1 series models (OpenAI et al., 2024) using large-scale reinforcement learning (RL).

DeepSeek further advanced test-time scaling by introducing the DeepSeek-R1 (DeepSeek-AI et al., 2025), successfully achieving performance comparable to OpenAI's o1 series. Prior approaches in inference test-time compute have included majority voting methods and external reward-based best-of-N strategies (Levi, 2024; Diao et al., 2024). Unlike repetitive sampling, sequential expansion approaches enable models to iteratively refine attempts based on prior outcomes. Many researchers have attempted to replicate or extend their methods, employing Monte Carlo Tree Search (MCTS) (Zhou et al., 2024; Choi et al., 2023), multi-agent approaches (Qin et al., 2024; Li et al., 2025), some work based on Process Reward Model(PRM) (Wang et al., 2024; Lightman et al., 2023).

Despite these successes, existing methods exhibit critical limitations. PRM face challenges such as defining fine-grained reasoning steps clearly, verifying intermediate reasoning correctness, and mitigating reward hacking (Amodei et al., 2016; Langosco et al., 2023), making automated labeling challenging and manual labeling impractical for scaling. Similarly, MCTS methods encounter difficulties due to vast search spaces, often causing models to become trapped in local optima, and depend heavily on sophisticated scoring models that are challenging to train (DeepSeek-AI et al., 2025).

Addressing these issues, DeepSeek introduced a rule-based reward system combined with large-scale reinforcement learning (RL), enabling clearer guidance and promoting model self-reflection and deeper reasoning (DeepSeek-AI et al., 2025). However, consistently identifying optimal reasoning paths remains challenging.

Inspired by human cognitive behaviors, we propose a novel test-time scaling strategy named **Multi-round Thinking**. This method allows the model to iteratively reconsider previous answers independently, using only the final answer from previous rounds as input prompts, discarding prior reasoning steps. This approach

parallels human cognitive processes, breaking cognitive inertia and enabling the model to correct entrenched reasoning errors.

Our experimental results demonstrate the effectiveness of this intuitive approach. For example, using the DeepSeek-R1 model (DeepSeek-AI et al., 2025), performance improvements were observed across multiple benchmarks: onAIME 2024 (MAA, 2024), pass@1 increased from 79.7% (Round 1) to 82.0% (Round 2); on GPQA-Diamond (Rein et al., 2023), it rose from 74.0% to 74.8%; and on LiveCodeBench (Jain et al., 2024), performance improved from 65.3% to 67.1%. These findings underscore the substantial potential of iterative thinking for further exploiting the benefits of test-time scaling.

2   Approach

We introduce a novel **Multi-round Thinking** approach designed to significantly enhance reasoning capabilities in large language models (LLMs). In contrast to traditional single-step reasoning methods, our approach iteratively refines answers through multiple rounds of inference. Each round takes the answer from the previous iteration (without intermediate reasoning steps) as part of a new input prompt, encouraging independent reconsideration and correction. This iterative process helps models avoid cognitive inertia, analogous to human strategies in overcoming entrenched errors in reasoning.

The Multi-round Thinking methodology operates explicitly as follows:

Given an original user prompt P_{user}, the inference and refinement process proceeds iteratively:

Initial Round (Round 1): The language model receives the initial prompt and generates the first round of reasoning and final answer:

$$M(P_{user}) \rightarrow \{Thinking_1, Answer_1\} \tag{1}$$

Subsequent Rounds (Round n, $n \geq 2$): In each subsequent inference round, intermediate reasoning traces ($Thinking_{n-1}$) from the previous iteration are discarded, retaining only the final answer ($Answer_{n-1}$). The prompt for the next round is constructed by concatenating the original user prompt and the previously obtained answer:

$$P_n = P_{user} \oplus Answer_{n-1} \tag{2}$$

The model independently reevaluates the newly formed prompt and produces an updated reasoning trace and refined answer:

$$M(P_n) \rightarrow \{Thinking_n, Answer_n\} \tag{3}$$

This iterative refinement cycle can be formally represented as follows:

$$P_1 = P_{user}, \quad M(P_1) \rightarrow \{Thinking_1, Answer_1\} \tag{4}$$

$$P_2 = P_{user} \oplus Answer_1, \quad M(P_2) \rightarrow \{Thinking_2, Answer_2\} \tag{5}$$

$$P_3 = P_{user} \oplus Answer_2, \quad M(P_3) \rightarrow \{Thinking_3, Answer_3\} \tag{6}$$

$$\vdots \tag{7}$$

$$P_n = P_{user} \oplus Answer_{n-1}, \quad M(P_n) \rightarrow \{Thinking_n, Answer_n\} \tag{8}$$

In these equations, $\oplus$ denotes the textual concatenation operation used to form the iterative prompts. Through this repeated refinement procedure, the model is encouraged to reconsider previous conclusions independently, effectively minimizing cognitive inertia and systematically improving the quality of reasoning outcomes.

Specifically, for a given question prompt P, we first use a reasoning model to answer the question, producing a thought process T and an answer A. Then, we concatenate the original question prompt P and the answer A using the following prompt:

```
Original question prompt
The assistant's previous answer is:  <answer> last round answer </answer>, and please
re-answer.
```

We then send this prompt to the large model again to generate a new answer. Using this method, we can obtain multi-turn responses.

3 Experiments

3.1 Evaluation

3.1.1 Benchmark

We evaluated the reasoning ability of the model using LiveCodeBench (Jain et al., 2024), GPQA-Diamond (Rein et al., 2023),AIME 2024 (MAA, 2024), and MATH-500 (Lightman et al., 2023). These benchmarks span multiple fields and difficulty levels, enabling a thorough assessment of the model's reasoning performance across diverse scenarios.

3.1.2 Evaluation Methodology

We standardized the evaluation conditions by setting the maximum generation length at 32,768 tokens. For benchmarks requiring stochastic sampling, we uniformly set the temperature to 0.6 and the top-p value to 0.95. Specifically, for AIME 2024 (MAA, 2024), we generated 32 samples per query to calculate pass@1 accuracy. For LiveCodeBench (Jain et al., 2024) and GPQA-Diamond (Rein et al., 2023), we generated 8 samples per query to estimate pass@1. For MATH-500 (Lightman et al., 2023), we generated 4 responses per query, also to estimate pass@1 accuracy. The primary evaluation metric adopted was the global average accuracy across all benchmarks.

3.2 Results and Analysis

3.2.1 Overall Results of Multi-round Thinking

Experimental results comparing initial (Round 1) and Multi-round Thinking (Round 2) performance are summarized in Table 1.

Table 1: Model Performance Comparison (pass@1 accuracy) Between Single-round (Round 1) and Multi-round Thinking (Round 2-4) Across Different Benchmarks

Model	Round	AIME 2024 pass@1	MATH500 pass@1	GPQA-Diamond pass@1	LiveCodeBench pass@1	Average
Deepseek-R1	1	79.7	97.6	74.0	65.3	79.2
	2	**82.0**	**97.6**	**74.8**	**67.1**	**80.4**
QwQ-32B	1	80.3	97.2	65.9	63.0	76.6
	2	82.1	97.8	67.2	64.7	78.0
	3	82.8	97.8	67.5	65.2	78.3
	4	**83.1**	**97.7**	**68.1**	**66.0**	**78.7**
DeepSeek-R1-Distill-Qwen-32B	1	72.0	96.0	60.1	57.0	71.3
	2	**75.1**	**96.3**	**61.3**	**57.6**	**72.6**
DeepSeek-R1-Distill-Qwen-7B	1	56.9	93.4	49.2	35.0	58.6
	2	**58.4**	**93.9**	**49.4**	**36.7**	**59.6**
AM-Distill-Qwen-32B	1	72.8	96.2	62.3	58.3	72.4
	2	**76.7**	**97.2**	**62.8**	**60.2**	**74.2**

Experimental results consistently show that our proposed Multi-round Thinking method effectively enhances reasoning performance across diverse benchmarks. As detailed in Table 1, each evaluated model showed notable improvement when transitioning from Round 1 to Round 2 reasoning.

Specifically, for the **Deepseek-R1** model, accuracy improved from 79.7% to 82.0% on theAIME 2024 benchmark, remained consistently high at 97.6% on MATH-500, increased from 74.0% to 74.8% on GPQA-Diamond, and improved from 65.3% to 67.1% on LiveCodeBench.

For the **QwQ-32B** model, notable gains were achieved, with accuracy rising from 80.3% to 82.1% onAIME 2024 (MAA, 2024), 97.2% to 97.8% on MATH-500, 63.0% to 64.7% on GPQA-Diamond, and 65.9% to 67.2% on LiveCodeBench.

Further, we evaluated our self-trained **AM-Distill-Qwen-32B** model, a 32B model built upon the Qwen2.5-32B (Qwen, 2024) architecture and trained using distilled data from the DeepSeek-R1 model (refer to (Zhao et al., 2025) for distillation details). Experimental results demonstrated robust performance improvements with Multi-round Thinking: accuracy increased from 72.8% to 76.7% on AIME 2024, from 96.2% to 97.2% on MATH-500, from 62.3% to 62.8% on GPQA-Diamond, and from 58.3% to 60.2% on LiveCodeBench.

Building upon this, we further examine the performance trajectory of **QwQ-32B** across four rounds of iterative thinking, as visualized in Figure 1. The model exhibits a clear and steady upward trend across all benchmarks.

From Round 1 to Round 4, QwQ-32B's performance on **AIME 2024** improves from 80.3% to 83.1%, indicating enhanced capability in competition-level mathematical reasoning. On the **MATH-500** dataset, performance remains consistently high, fluctuating slightly within the 97.2%–97.8% range.

Substantial gains are observed on reasoning-heavy benchmarks like **GPQA-Diamond**, where accuracy increases from 65.9% to 68.1% over four rounds. Similarly, **LiveCodeBench** scores rise from 63.0% to 65.9%, reflecting a notable enhancement in code understanding and generation tasks.

These empirical results highlight the consistent advantage provided by the Multi-round Thinking methodology, underscoring its efficacy in iteratively refining reasoning processes, correcting earlier mistakes, and substantially boosting model performance across challenging reasoning tasks.

In summary, our results strongly indicate that Multi-round Thinking consistently improves the reasoning performance of LLMs across various tasks. Particularly notable is its effectiveness on tasks demanding complex, iterative reasoning such as mathematics competitions and coding benchmarks. Moreover, the incremental and sustained improvements observed over multiple rounds underscore the robustness of this simple yet effective test-time scaling strategy. These findings suggest that Multi-round Thinking offers an efficient pathway to enhance model accuracy without additional training overhead, thus highlighting its practical value for real-world deployment and opening promising avenues for future research in test-time scaling methods.

3.2.2   Analysis of Word Frequency Changes

To better understand how the model's reasoning behavior evolves through multi-round thinking, we conduct a lexical analysis focusing on four discourse markers: but, wait, maybe, and therefore. These words serve as linguistic signals for hesitation (but, wait, maybe) or decisiveness (therefore), and tracking their usage reveals insights into the model's confidence and reasoning dynamics.

Figure 3 presents the overall average usage frequency of these keywords across all AIME 2024 test samples. From Round 1 to Round 2, we observe consistent declines in the frequency of hesitation-related words. Specifically, but decreases from 68.3 to 44.8, wait from 67.9 to 51.0, and maybe from 23.7 to 15.8. Even therefore, although a more conclusive word, experiences a drop from 43.8 to 29.5, but the relative reduction is smaller. This suggests that, overall, the model adopts more concise and assertive phrasing in Round 2 responses.

To analyze this shift in greater detail, Figure 4 breaks down average keyword usage by answer trajectory: Incorrect→Correct (I-C), Incorrect→Incorrect (I-I), Correct→Correct (C-C), and Correct→Incorrect (C-I). In most groups, there is a significant drop in but, wait, and maybe from Round 1 to Round 2, reinforcing the observation that models tend to suppress uncertain or self-interruptive phrasing after one round of reflection. For example, in the I-I group, but drops from 145.2 to 106.8 and wait from 131.4 to 110.8, indicating that even when the model fails in both rounds, it still shifts toward more direct expression.
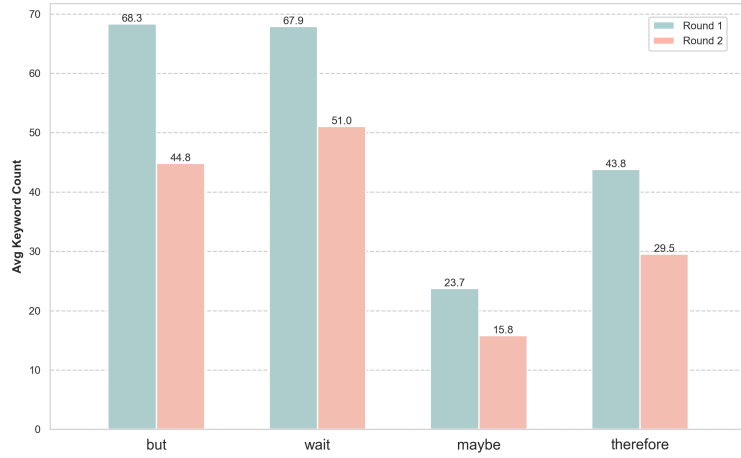
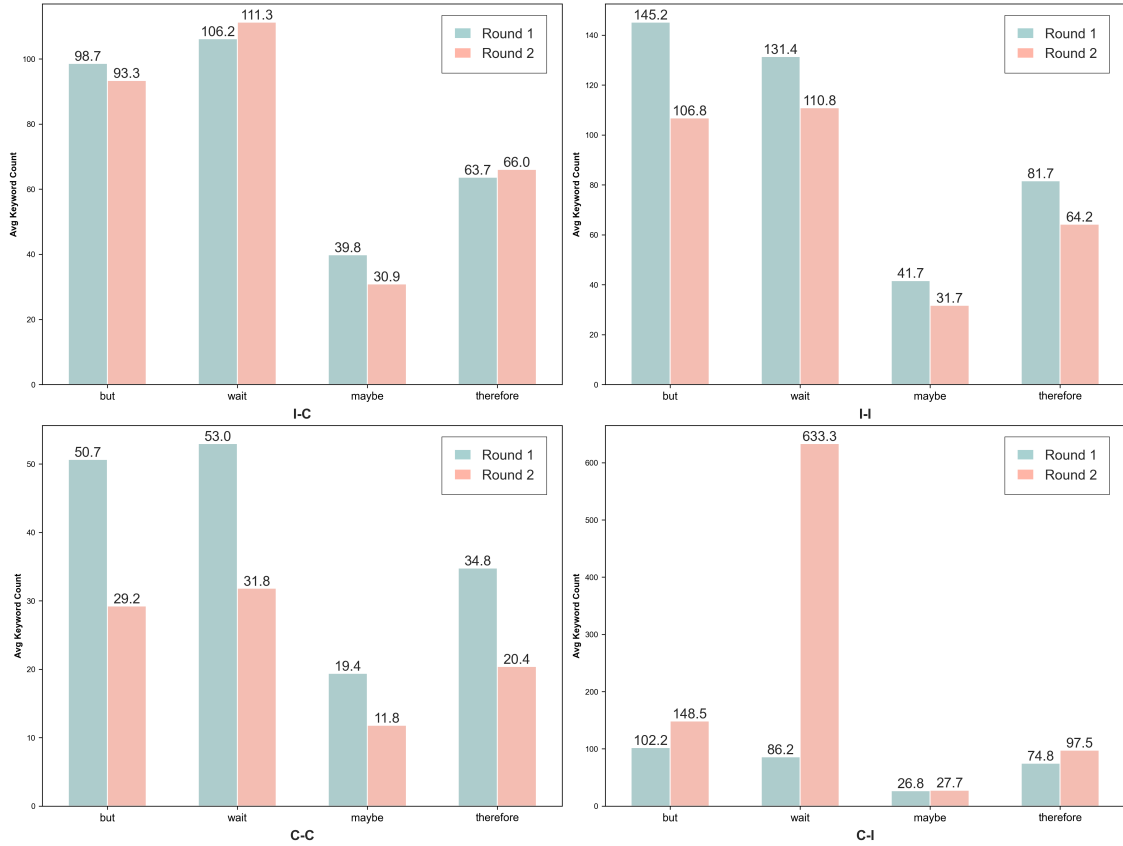Figure 3: Overall change in word frequency across all AIME 2024 examples.

Figure 4: Changes in average word frequency across different reasoning trajectories. Each subplot shows the average frequency of four indicative words — *but*, *wait*, *maybe*, and *therefore* — in Round 1 vs. Round 2, grouped by response type: I-C (Incorrect → Correct), I-I (Incorrect → Incorrect), C-C (Correct → Correct), and C-I (Correct → Incorrect).

Interestingly, in the I-C group where the model corrects its earlier error, we observe an increase in the use of wait and therefore. This suggests a more thoughtful and deliberate step-by-step reanalysis process. The rise in therefore (from 63.7 to 66.0) also reflects the model's increased confidence in arriving at the correct conclusion.

Together, these patterns suggest that multi-round thinking helps the model become more confident, fluent, and decisive in its responses—reducing hedging and strengthening clarity.

3.2.3 Analysis of Response Length

Table 2: Analysis of response length across multiple reasoning rounds for QwQ-32B, demonstrating decreasing token lengths as the model iteratively refines answers.

Model	Round	AIME 2024	MATH500	GPQA-Diamond	LiveCodeBench	Average
QwQ-32B	1	13566.1	8489.9	13860.5	4473.0	10097.4
	2	9607.9	5540.4	11043.9	3200.9	7348.3
	3	8630.0	5287.7	10368.0	3012.7	6824.6
	4	8654.8	4948.0	9674.5	2920.8	6549.5

We analyzed the generation lengths of the model across different inference rounds, as shown in Table 2.

As the number of inference rounds increases, the generation length tends to decrease. Moreover, there is a correlation between performance improvement and the reduction in generation length—the greater the performance gain, the more significant the reduction. For example, from Round 1 to Round 2, the average score improves by 1.4 points while the average generation length decreases by 2749.1 tokens; from Round 2 to Round 3, there is minimal improvement in performance, while the average generation length decreases by only 675.9 tokens.
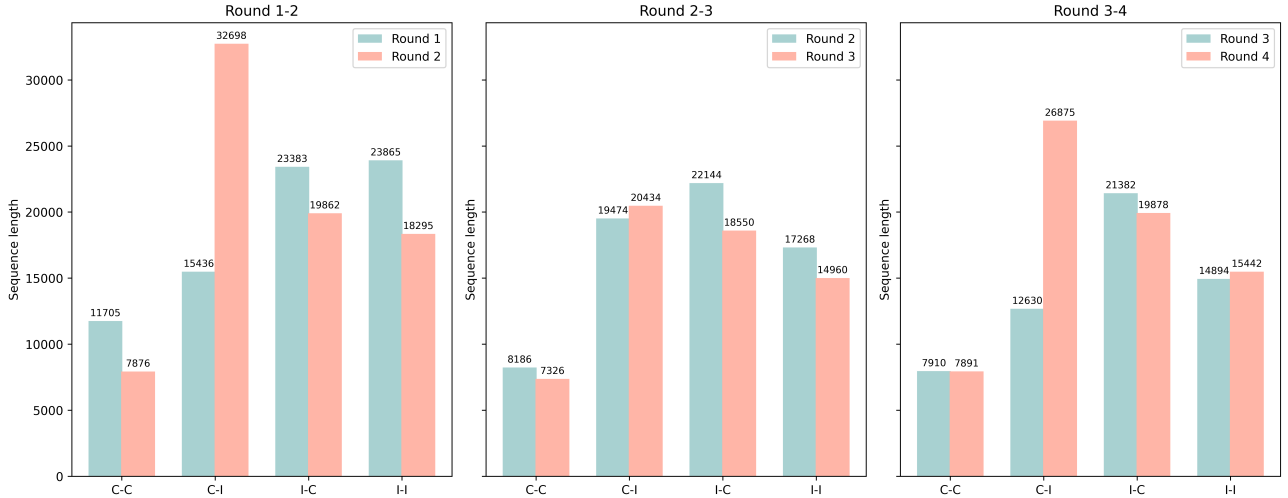

Figure 5: Changes in response length across reasoning rounds on theAIME 2024 dataset (QwQ-32B model). Labels represent the correctness trajectory from Round 1 to Round 2: "C" = Correct, "I" = Incorrect. For example, "C–I" indicates responses that were correct initially but became incorrect in the next round.

As shown in Figure 5, we analyzed the changes in response lengths for the same questions across different inference rounds. We found that when the model answered correctly in the previous round but incorrectly

in the current round, the inference length increased significantly. This trend is consistent with the word frequency analysis in Figure 3, where the frequency of the word "wait" rises notably, suggesting increased uncertainty in the model's reasoning. In contrast, when the model answers correctly in both the previous and current rounds, it becomes more confident, leading to a substantial reduction in inference length.

3.3 A Preliminary Experiment with Supervised Fine-tuning (SFT)

To further enhance the robustness of Multi-round Thinking, we explored combining it with supervised fine-tuning (SFT). The key idea was to reduce error propagation from earlier reasoning rounds by explicitly training the model to rectify previously incorrect answers.

Specifically, our supervised fine-tuning process involved the following steps:

- **Data selection**: We selected challenging mathematical and programming tasks from open datasets, ensuring each could be independently verified.

- **Data generation**: We employed the DeepSeek-R1 model iteratively on these tasks using Multi-round Thinking until a correct answer was verified, forming a dataset of approximately 100,000 examples.

- **Model training**: This dataset was used to fine-tune our AM-32B model in an initial round of supervised training.

Table 3: Model Performance with Average(pass@1)

Model	Round	AIME 2024	MATH-500	GPQA-Diamond	LiveCodeBench	Average
AM-32B	1	72.8	96.2	62.3	58.3	72.4
	2	**76.7**	**97.2**	**62.8**	**60.2**	**74.2**
	2(SFT)	75.9	97.0	63.0	57.9	73.5

After SFT, we conducted further experiments based on the AM-32B Round2 reasoning outputs. While this preliminary fine-tuning did not lead to performance improvements in our current evaluation (see Table 3), it opens up promising directions for future research in leveraging high-quality reasoning data to enhance Multi-round Thinking.

4   Discussion and Conclusion

In this study, we proposed Multi-round Thinking, a straightforward yet effective test-time scaling strategy designed to enhance the reasoning capabilities of large language models (LLMs). Inspired by human cognitive processes, this iterative approach allows models to refine their reasoning by independently reconsidering their previous answers, significantly mitigating cognitive inertia and correcting initial reasoning errors. Our extensive experiments demonstrated consistent and substantial improvements across challenging benchmarks, including AIME 2024, GPQA-Diamond, MATH-500, and LiveCodeBench. For instance, accuracy improved by more than 2 percentage points on complex mathematical competition tasks, underscoring the broad applicability and practical value of this approach.

Further analysis revealed that multi-round reasoning not only improved accuracy but also made the models' reasoning more concise and confident. Specifically, we observed a reduction in uncertainty markers (such as "but," "wait," and "maybe") and shorter responses, reflecting increased model clarity and decisiveness in reasoning. These linguistic insights indicate that iterative thinking aligns closely with human cognitive patterns, enhancing the transparency and interpretability of LLM behaviors.

While preliminary experiments integrating supervised fine-tuning (SFT) did not immediately yield additional improvements, they highlighted crucial considerations for future research—particularly regarding the

quality of training data and fine-tuning strategies tailored explicitly for iterative reasoning. Exploring these directions further promises significant theoretical and practical benefits, potentially unlocking even greater reasoning capabilities in LLMs.

In practical product applications, adopting a "think twice" approach can conveniently incorporate the first-round response as part of the thinking process itself, effectively realizing performance gains. However, this inevitably introduces additional waiting time during the thinking phase.

In summary, Multi-round Thinking represents a practical, efficient, and universally applicable method for improving LLM reasoning without additional training overhead. This research opens valuable pathways for future exploration and offers immediate utility for both academia and industry in the ongoing quest for more robust, reliable, and explainable AI reasoning.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety, 2016. URL https://arxiv.org/abs/1606.06565.

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

Sehyun Choi, Tianqing Fang, Zhaowei Wang, and Yangqiu Song. Kcts: Knowledge-constrained tree search decoding with token-level hallucination detection, 2023. URL https://arxiv.org/abs/2310.09044.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Shizhe Diao, Pengcheng Wang, Yong Lin, Rui Pan, Xiang Liu, and Tong Zhang. Active prompting with chain-of-thought for large language models, 2024. URL https://arxiv.org/abs/2302.12246.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Lauro Langosco, Jack Koch, Lee Sharkey, Jacob Pfau, Laurent Orseau, and David Krueger. Goal misgeneralization in deep reinforcement learning, 2023. URL https://arxiv.org/abs/2105.14111.

Noam Levi. A simple model of inference scaling laws, 2024. URL https://arxiv.org/abs/2410.16377.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models, 2025. URL https://arxiv.org/abs/2501.05366.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

MAA. American invitational mathematics examination - aime. https://maa.org/math-competitions/american-invitational-mathematics-examination-aime, feb 2024. Accessed in February 2024, from American Invitational Mathematics Examination - AIME 2024.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393.

OpenAI. Learning to reason with llms, 2024. URL https://openai.com/index/learning-to-reason-with-llms/.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon,

Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

Yiwei Qin, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan, Hector Liu, Yuanzhi Li, and Pengfei Liu. O1 replication journey: A strategic progress report – part 1, 2024. URL https://arxiv.org/abs/2410.18982.

Qwen. Team qwen2.5: A party of foundation models, September 2024. URL https://qwenlm.github.io/blog/qwen2.5/.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023. URL https://arxiv.org/abs/2311.12022.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations, 2024. URL https://arxiv.org/abs/2312.08935.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models, 2025. URL https://arxiv.org/abs/2408.00724.

Wenkai Yang, Shuming Ma, Yankai Lin, and Furu Wei. Towards thinking-optimal scaling of test-time compute for llm reasoning, 2025. URL https://arxiv.org/abs/2502.18080.

Han Zhao, Haotian Wang, Yiping Peng, Sitong Zhao, Xiaoyu Tian, Shuaiting Chen, Yunjie Ji, and Xiangang Li. 1.4 million open-source distilled reasoning dataset to empower large language model traning, 2025. URL https://github.com/a-m-team/a-m-models/blob/main/docs/AM-DeepSeek-R1-Distilled-Dataset.pdf.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models, 2024. URL https://arxiv.org/abs/2310.04406.
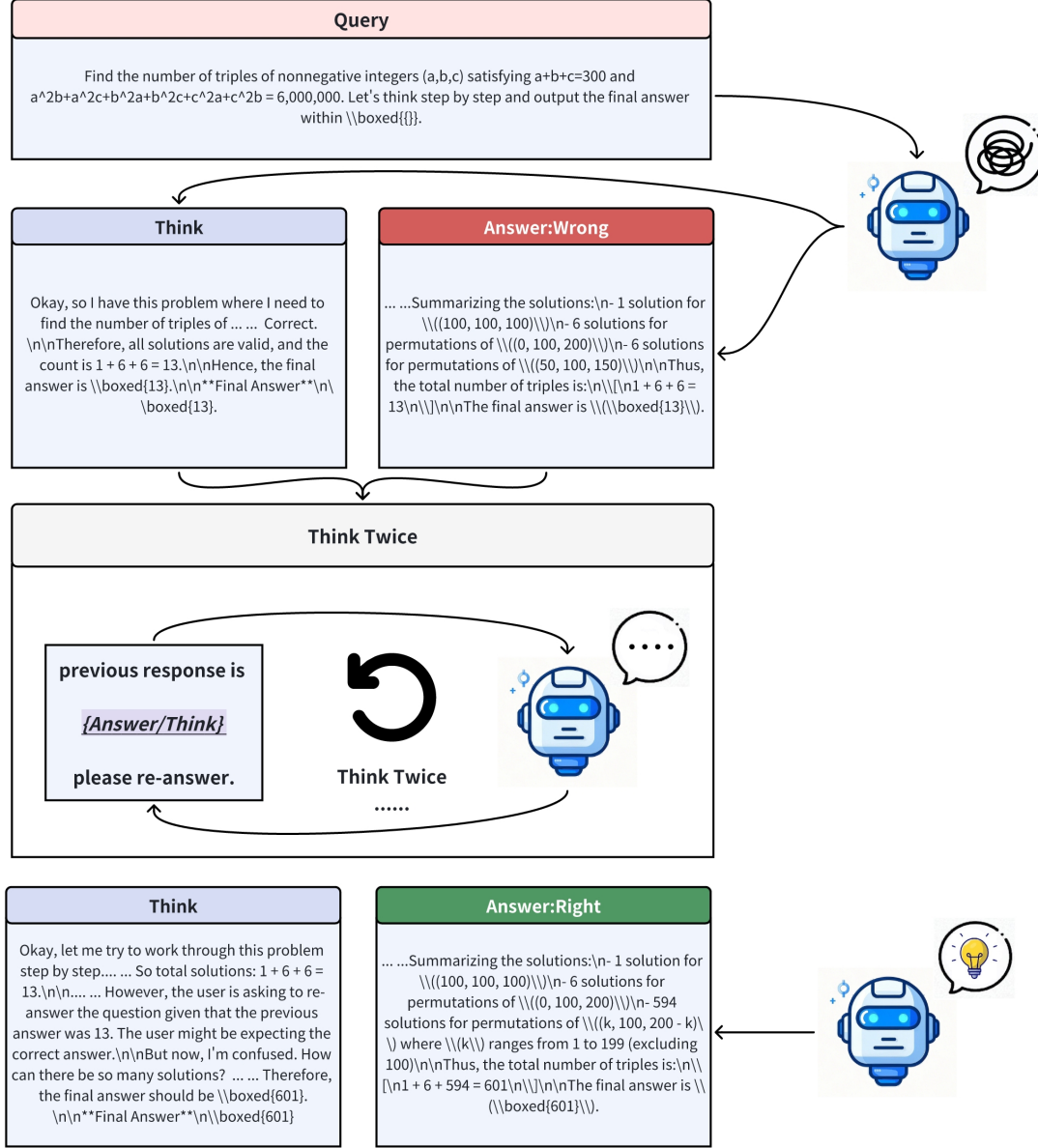
A   Example

Figure 6: llustration of the "Think Twice" Strategy in Multi-round Reasoning.

The model first provides an incorrect answer by following its initial reasoning chain.Upon invoking the `Think Twice` mechanism, it is explicitly prompted to reassess its prior response. The model then revisits its reasoning, identifies the over-simplified solution space, and produces a corrected answer with a significantly expanded and accurate enumeration. This process highlights the effectiveness of forcing self-reflection to catch subtle counting mistakes.