Wrong-of-Thought: An Integrated Reasoning Framework with Multi-Perspective Verification and Wrong Information

Yongheng Zhang[1] **Qiguang Chen**[2] **Jingxuan Zhou**[1] **Peng Wang**[1]
Jiasheng Si[3] **Jin Wang**[4] **Wenpeng Lu**[3] **Libo Qin**[1*]
[1]School of Computer Science and Engineering, Central South University, China
[2]Research Center for SCIR, Harbin Institute of Technology, Harbin, China
[3]Key Laboratory of Computing Power Network and Information Security, Ministry of Education
Qilu University of Technology (Shandong Academy of Sciences), China
[4]Yunnan University, Kunming, China

Abstract

Chain-of-Thought (CoT) has become a vital technique for enhancing the performance of Large Language Models (LLMs), attracting increasing attention from researchers. One stream of approaches focuses on the iterative enhancement of LLMs by continuously verifying and refining their reasoning outputs for desired quality. Despite its impressive results, this paradigm faces two critical issues: (1) *Single verification method*: The current paradigm relies solely on a single verification method. (2) *Wrong Information Ignorance*: The traditional paradigm directly ignores wrong information during reasoning and refines the logic paths from scratch each time. To address these challenges, we propose Wrong-of-Thought (`WoT`), which includes two core modules: (1) *Multi-Perspective Verification*: A multi-perspective verification method for accurately refining the reasoning process and result, and (2) *Wrong Information Utilization*: Utilizing wrong information to alert LLMs and reduce the probability of LLMs making same mistakes. Experiments on 8 popular datasets and 5 LLMs demonstrate that `WoT` surpasses all previous baselines. In addition, `WoT` exhibits powerful capabilities in difficult computation tasks.

1 Introduction

Failure is the mother of success.

– Chinese Idiom

In recent years, large language models (LLMs) have made significant advancements in a series of natural language processing tasks (Achiam et al., 2023; Touvron et al., 2023). Additionally, with the emergence of Chain-of-Thought (CoT) (Wei et al., 2022), the performance of LLMs has been further unlocked by guiding them through step-by-step reasoning (Liu et al., 2023a; Qin et al., 2023).

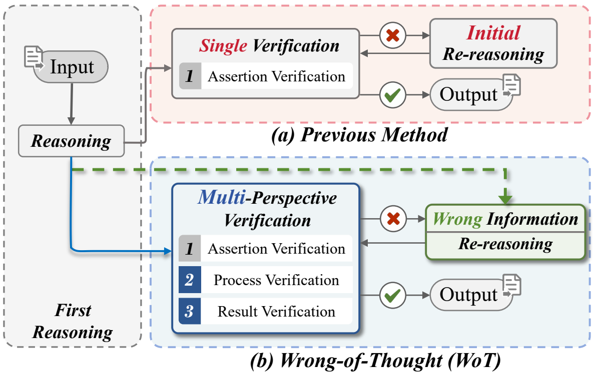


Figure 1: Previous multi-thoughts integration methods (a) vs. Wrong-of-Thought (b). Previous methods only used a ***Single*** *Verification* and did not utilize the wrong information. In contrast, `WoT` offers ***Multi****-Perspective Verification* and utilizes ***Wrong*** *Information*.

A common category of CoT research focuses on iteratively enhancing LLMs by continuously re-verifying and refining corresponding reasoning outputs to achieve better quality. Madaan et al. (2023) demonstrate this approach by prompting the model to self-verify the results and provide feedback on previously generated drafts, producing better outputs. Similarly, Chen et al. (2023) improve code debugging by leveraging external program execution results and model explanation code. When examining methodologies to guide the rethinking of models, Zheng et al. (2024) emphasize the reuse of previously generated answers. Meanwhile, Qi et al. (2023) introduce a problem-solving framework inspired by human divide-and-conquer strategies, which incorporates self-questioning and recursive thinking processes. Building upon this, Liu et al. (2023b) propose XoT, shown as Figure 1 (a), which integrates multiple reasoning paths with multiple logical modes. Specifically, they generate the rationale in program format and apply a single verification method to check the correctness of the reasoning. If errors are detected, the LLMs are instructed to switch to another reasoning thought

* Corresponding Author.

Findings of the Association for Computational Linguistics: EMNLP 2024, pages 6644–6653
November 12-16, 2024

and start the reasoning process from scratch. Despite achieving impressive results, they still face two significant challenges:

(1) *Single verification method*: They rely solely on the single verification method like basic syntax assertions, resulting in errors that fail to fully evaluate and validate the reasoning of models. This approach leads to suboptimal verification accuracy, significantly impeding overall reasoning performance.

(2) *Wrong Information Ignorance*: Once the error is detected, they typically disregard wrong information and re-generate the reasoning from scratch. However, it also loses a large amount of feedback signals brought by error information, which is often considered very important in model understanding (Zhang et al., 2024a; Tong et al., 2024; Chen et al., 2024b).

Motivated by this, we introduce the Wrong-of-Thought (`WoT`) framework, as illustrated in Figure 1 (b). To address the first challenge, we introduce *Multi-Perspective Verification*, which incorporates two additional explicit verification methods, mirroring human problem-solving processes. First, it ensures the variables in equations or code match the information provided in the question. Second, it resolves again the question to check for consistency in the results. We instruct LLMs to integrate these two perspectives to enhance solution verification. To address the second challenge, we introduce *Wrong Information Utilization*, which utilizes previous wrong reasoning information to guide LLMs in avoiding similar mistakes. By referencing past mistakes, LLMs can enhance their reasoning performance and minimize repetitive errors.

Experiments are conducted on 8 datasets and 5 LLMs. The results indicate that the `WoT` performs exceptionally well across all benchmark tests, surpassing all existing baselines. Furthermore, in-depth analytical experiments demonstrate that `WoT` excels at difficult computational tasks.

The key contributions of this work are:

(1) We first point out two main drawbacks in iterative reasoning, which lie in the monotonous verification perspective and the ignorance of wrong information feedback for ultimate limited logical improvement.

(2) We introduce Wrong-of-Thought (`WoT`) to solve these drawbacks, which mainly consists of two modules: *Multi-Perspective Verification* and *Wrong Information Utilization*. These modules enable accurate verification and effective utilization of wrong information.

(3) Our experiments on 8 datasets and 5 LLMs have shown that `WoT` achieves superior performance. In addition, `WoT` demonstrates strong problem-solving abilities in questions involving difficult mathematical reasoning.

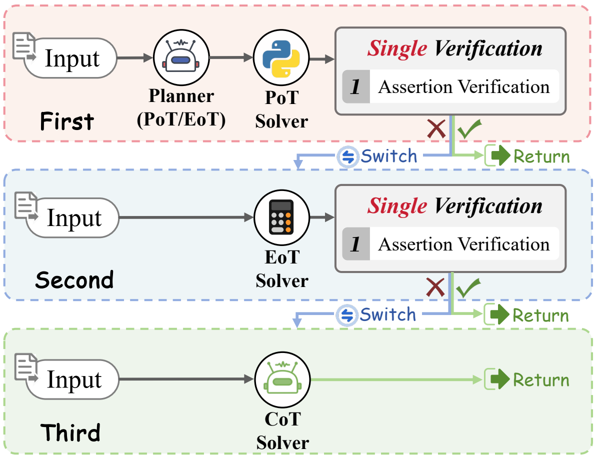


Figure 2: XoT Framework. First, select a reasoning method, either PoT or EoT and then apply assertion verification to make a judgment. If the reasoning is found to be incorrect, switch to the alternative method and restart the reasoning. Verify again, and if the verification is correct, return the answer. If the reasoning reaches the third step, utilize CoT reasoning as the answer.

All code will be open-sourced and publicly available at https://github.com/BRZ911/Wrong-of-Thought.

2 Preliminary

This section introduces the framework that mainstream integrated multiple reasoning thoughts, iteratively enhancing LLMs by continuously re-verifying and refining corresponding reasoning. XoT (Liu et al., 2023b), as shown in Figure 2, is an integrated reasoning framework that combines three reasoning modes: Program-of-Thought (PoT) (Chen et al., 2022), Equation-of-Thought (EoT) (Liu et al., 2023b), and Chain-of-Thought (CoT) (Wei et al., 2022). PoT enables LLMs to generate Python code and then uses the external Python executor to run the results. EoT involves LLMs generating mathematical equations, which are then solved using an external calculator. CoT is a technique that guides LLMs to reason step-by-step.

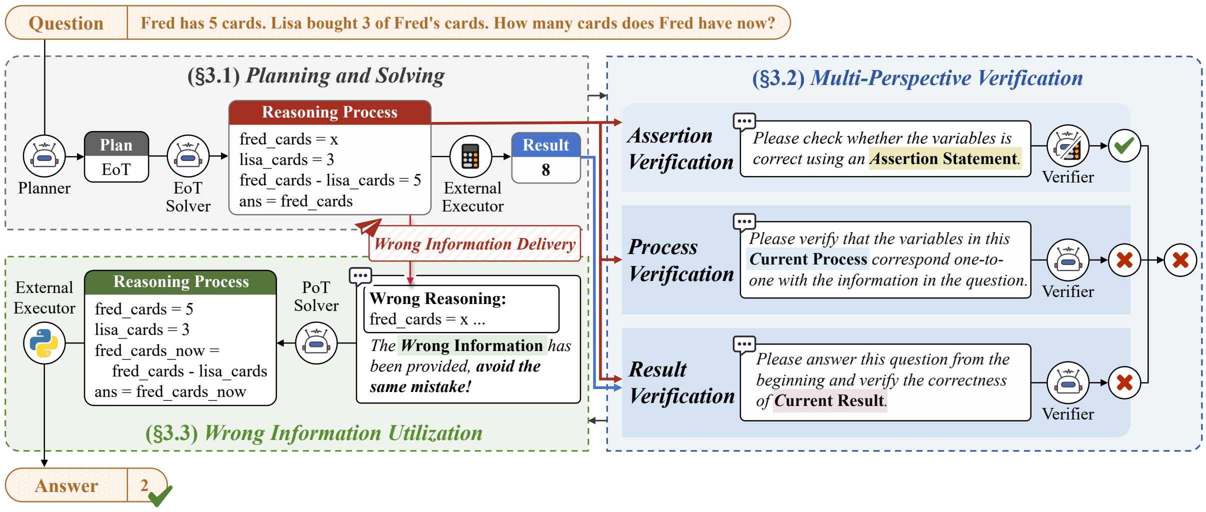


Figure 3: Overview of the Wrong-of-Thought (WoT) framework, incorporating three core modules: *Planning and Solving* (§3.1), ***Multi-Perspective Verification*** (§3.2), and ***Wrong Information Utilization*** (§3.3).

In the XoT framework, the **First** step involves initiating the reasoning plan and selecting a reasoning method from either EoT or PoT to perform the reasoning. Once the reasoning process is completed, the result is computed through an external executor. The answer is then verified using an assertion verification. If the reasoning result is determined to be correct, the answer is returned. If the initial reasoning is deemed incorrect and abandoned, the **Second** step is to switch to an alternative reasoning mode and restart the process. After obtaining and verifying the new reasoning answer, if it is still incorrect, the **Third** step is to directly use CoT reasoning as the final answer.

3 Wrong-of-Thought

This section introduces Wrong-of-Thought (WoT). The content is divided into three parts: *Planning and Solving* (§3.1), *Multi-Perspective Verification* (§3.2), and *Wrong Information Utilization* (§3.3).

3.1 Planning and Solving

Following XoT (Liu et al., 2023b), as shown in Figure 3 (§3.1), initially, a planner selects a reasoning method from either EoT or PoT based on the inputted question. After the Solver module generates the reasoning process, an external executor computes the result, yielding a preliminary solution. The next step is to validate the current solution.

3.2 Multi-Perspective Verification

To address the challenge of verification methods being singular and significantly hindering the overall performance, we propose a *Multi-Perspective Verification* (MPV), as shown in Figure 3 (§3.2). Specifically, *Multi-Perspective Verification* is applicable to the reasoning verification of EoT and PoT, which includes the following three aspects:

(1) **Assertion Verification**: We adopt the verification method from the XoT (Liu et al., 2023b). We use LLMs to identify the intermediate variables in the solution and format them as Assertion Statements. These assertion statements are then executed using external tools to obtain the verification results.

(2) **Process Verification**: For process verification, we provide the LLMs only with the Current Process, excluding the computed results. We ask the LLMs to recheck each step of the current reasoning process to ensure that the variables in the solution equations or code correspond one-to-one with the question information, explicitly demonstrating the verification reasoning process.

(3) **Result Verification**: In the results verification phase, we provide the LLMs with both the current reasoning process and the computed results. We instruct the LLMs to recheck the Current Result by re-solving the problem. If the result passes re-verification, the LLMs output "right"; otherwise, they output "error". This explicitly demonstrates the verification reasoning results.

To enhance the robustness of our verification, we employ a voting mechanism to select the judgments that exhibit higher consistency across different verification perspectives V_i. These consistent judgments are then used as the final MVP results $\hat{V}$ for the output R of the reasoning method M_i. The verification can be formalized as follows:

$$\hat{V} = \underset{V_t \in \mathcal{V}}{\text{argmax}} \sum_{t=1}^{N} \sum_{R \in M_i} \mathbb{1}(V_t = R), \quad (1)$$

where V_t represents verification methods, $\mathcal{V}$ represents the set of the three verification methods, R represents the output using the reasoning method M_i, and $\mathbb{1}(V_t = R)$ returns 1 if the verification method V_t matches output R, and 0 otherwise.

3.3 Wrong Information Utilization

To address the issue of previous methods ignoring wrong information, we propose *Wrong Information Utilization* (WIU), as shown in Figure 3 (§3.3). Specifically, after the previous solution is validated and determined to be wrong, we incorporate the prior Wrong Information within the context of the current solution method. This guides the LLMs to avoid repeating the same mistakes. Formally, the reasoning for the question Q after utilizing wrong reasoning information can be expressed by the following formula:

$$\hat{R} = \underset{R \in M_i}{\text{argmax}} \, P(R|Q, I, \text{WI}), \quad (2)$$

where $\hat{R}$ represents the final reasoning result. $P(R|Q, I, \text{WI})$ denotes the probability of generating the reasoning path R under the conditions of question Q, prompt I, and Wrong Information WI. R is a reasoning of the reasoning method M_i.

After obtaining the reasoning results, we use the *Multi-Perspective Verification* to make a judgment. If the judgment is correct, the answer is returned directly. If the judgment is wrong, following XoT, we proceed to the third step, where the errors from this step and the previous step will be used as wrong examples for CoT reasoning.

4 Experiments

4.1 Experimental Setting

We conduct experiments on eight widely used comprehensive datasets, including GSM8K (Cobbe et al., 2021), GSM-Hard (Gao et al., 2023), Algebra (He-Yueya et al., 2023), MultiArith (Roy and Roth, 2015), SingleEQ (Koncel-Kedziorski et al., 2015), SingleOP (Roy et al., 2015), AddSub (Hosseini et al., 2014), and SVAMP (Patel et al., 2021). The effectiveness of the `WoT` framework was validated on these challenging benchmarks.

Additionally, we select the single reasoning methods CoT (Wei et al., 2022), PoT (Chen et al., 2022), and EoT (Liu et al., 2023b), as well as the ensemble method XoT (Liu et al., 2023b), as baselines. The verification process was conducted on a comprehensive set of five LLMs. Among these, three are open-source LLMs: Mistral-7B-Instruct (Jiang et al., 2023), Qwen-7B-Chat (Bai et al., 2023), and Qwen-14B-Chat (Bai et al., 2023). The other two LLMs are closed-source: Gemini-1.0-Pro (Team et al., 2023) and GPT-3.5-Turbo (OpenAI, 2022). These models were selected to provide a diverse representation of current advanced LLMs, both open and closed-source, ensuring a robust and comprehensive verification.

Following XoT (Liu et al., 2023b), all experiments used 8-shot correct examples as prompts. The experimental results were evaluated using Accuracy as the evaluation metric. The top-p and temperature parameters for all experiments were set to LLMs default parameters in the official model configuration, which are within the range of [0,1].

4.2 Main Results

The main experimental results are shown in Table 1. Based on the results, we can observe:

(1) `WoT` reaches superior performance. `WoT` surpasses all baselines, achieving superior performance on eight datasets, with an average improvement of 2.8% compared to XoT across five LLMs. This extensive experimental result demonstrates the effectiveness of the integration of *Multi-Perspective Verification* and *Wrong Information Utilization* in `WoT`, enhancing overall performance.

(2) `WoT` can also work on LLMs with smaller parameters. `WoT` achieves an average improvement of 4.2% and 2.3% on the smaller parameter open-source models, Mistral-7B-Instruct and Qwen1.5-7B-Chat, respectively, demonstrating robust performance. The ability of `WoT` to maintain high performance on models with fewer parameters highlights its potential for broad applicability in various practical scenarios, including those with limited computational resources.

(3) `WoT` demonstrates a powerful ability to solve difficult reasoning questions. `WoT` achieves an average performance on GSM-Hard that was 5.7% higher than the baselines on five LLMs, represent-

Method	GSM-hard	GSM8K	Algebra	MultiArith	SingleEQ	SingleOP	AddSub	SVAMP	Average
				Mistral-7B-Instruct (Jiang et al., 2023)					
CoT (Wei et al., 2022)	16.6	47.5	36.0	68.8	78.3	81.1	73.9	60.8	57.9
PoT (Chen et al., 2022)	30.8	45.0	28.4	72.8	75.8	64.4	74.7	56.5	56.0
EoT (Liu et al., 2023b)	16.1	22.3	27.0	25.0	31.1	33.6	29.1	23.5	26.0
XoT (Liu et al., 2023b)	26.2	52.8	46.8	77.8	86.6	85.4	80.0	67.9	65.5
Wrong-of-Thought	**36.7**	**54.6**	**50.5**	**80.8**	**88.0**	**87.9**	**88.9**	**70.0**	**69.7**
				Qwen-7B-Chat (Bai et al., 2023)					
CoT (Wei et al., 2022)	18.6	52.8	43.7	83.2	87.4	83.1	80.5	70.7	65.0
PoT (Chen et al., 2022)	39.0	56.2	38.7	84.8	90.6	89.7	82.5	71.3	69.1
EoT (Liu et al., 2023b)	35.3	49.2	34.2	61.5	76.0	63.5	65.1	48.0	54.1
XoT (Liu et al., 2023b)	38.3	61.8	54.5	88.7	92.1	92.3	85.1	76.4	73.6
Wrong-of-Thought	**42.0**	**63.7**	**57.2**	**91.3**	**94.1**	**93.6**	**86.3**	**79.3**	**75.9**
				Qwen-14B-Chat (Bai et al., 2023)					
CoT (Wei et al., 2022)	31.0	63.4	56.8	89.8	88.0	85.4	85.3	80.8	72.6
PoT (Chen et al., 2022)	57.1	69.5	62.6	95.7	95.7	96.1	86.8	81.6	80.6
EoT (Liu et al., 2023b)	57.6	68.5	62.6	85.7	90.6	82.2	83.8	79.2	76.3
XoT (Liu et al., 2023b)	55.3	76.3	80.2	92.0	94.1	94.5	86.1	84.8	82.9
Wrong-of-Thought	**60.6**	**77.5**	**81.5**	**98.3**	**96.7**	**95.4**	**88.1**	**86.3**	**85.5**
				Gemini-1.0-Pro (Team et al., 2023)					
CoT (Wei et al., 2022)	45.6	81.9	81.5	94.8	96.1	94.7	92.9	83.0	83.8
PoT (Chen et al., 2022)	63.8	77.1	58.1	96.3	96.3	96.3	91.6	87.1	83.3
EoT (Liu et al., 2023b)	52.2	61.1	63.5	80.0	79.7	75.3	78.0	71.3	70.1
XoT (Liu et al., 2023b)	64.6	82.1	83.3	96.5	96.1	96.3	91.4	86.9	87.2
Wrong-of-Thought	**69.1**	**84.4**	**85.6**	**97.3**	**97.4**	**97.3**	**93.4**	**89.2**	**89.2**
				GPT-3.5-Turbo (OpenAI, 2022)					
CoT (Wei et al., 2022)	42.2	80.0	72.1	97.3	96.5	94.7	89.4	80.2	81.5
PoT (Chen et al., 2022)	70.3	77.4	81.5	97.8	98.6	94.3	88.9	79.2	86.0
EoT (Liu et al., 2023b)	53.4	64.0	70.3	84.8	61.4	68.5	70.1	58.9	66.4
XoT (Liu et al., 2023b)	71.3	83.6	84.7	97.8	97.6	94.5	89.4	83.0	87.7
Wrong-of-Thought	**76.2**	**85.2**	**89.6**	**99.0**	**99.0**	**96.1**	**93.2**	**86.7**	**90.6**

Table 1: Experimental results of Acc. (%) on eight datasets and five LLMs. **Bold** represents the best performance.

ing a significant improvement. The GSM-Hard dataset, a mathematical reasoning dataset where small numerical values are replaced with large ones (average result: $7.3e9$), demonstrates the strong performance of WoT in difficult reasoning tasks.

4.3 WoT Analysis

To gain a more profound understanding of WoT, we propose the following research questions based on experiments on GPT-3.5-Turbo (OpenAI, 2022):

(1) *Can Wrong Information Utilization lead to performance improvement?*

(2) *Can Multi-Perspective Verification lead to more accurate judgment results?*

(3) *Can WoT reduce the number of required reasoning steps?*

(4) *Why does WoT have strong capabilities in difficult mathematical reasoning?*

(5) *What is the intuition behind WoT?*

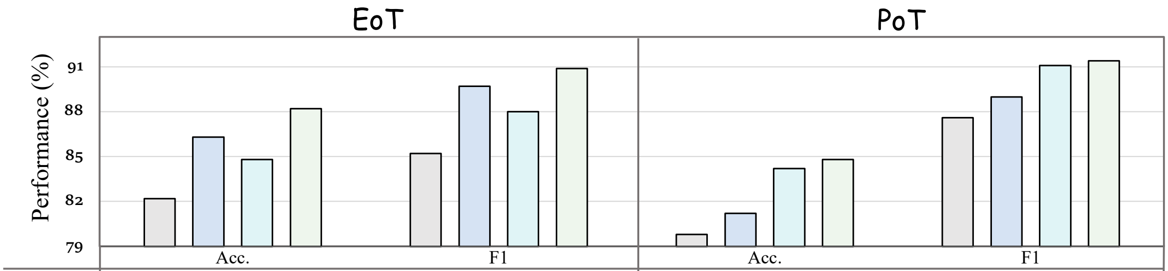
4.3.1 Answer 1: *Wrong Information Utilization* can boost performance

To intuitively verify the performance improvements brought by using wrong information, we select PoT, EoT, and CoT that utilized wrong information from the GSM8K dataset for evaluation. We compare their performance with and without wrong information. Additionally, we test the WoT performance without the *Wrong Information Utilization*. Due to the limitation within the WoT, EoT and PoT can only collect incorrect information once, resulting in a single wrong example. On the other hand, CoT can collect incorrect information up to two times, resulting in two wrong examples.

The results are shown in the Figure 5. After incorporating wrong information from the previous step, EoT and PoT improved by 8% and 8.9%, respectively. We can observe that CoT, which utilized additional wrong information from the previous two steps, improved by 13.1%. Furthermore, as shown in Table 2, the WoT framework without

Methods	GSM-hard	GSM8K	Algebra	MultiArith	SingleEQ	SingleOP	AddSub	SVAMP	AVG
Wrong-of-Though	76.2	85.2	89.6	99.0	99.0	96.1	93.2	86.7	90.6
w/o WIU	73.9	84.0	87.8	98.8	98.4	95.9	92.6	85.5	89.6 *(-1.0)*
w/o MPV	73.1	82.4	87.4	98.3	98.6	94.5	90.4	85.6	88.8 *(-1.8)*
w/o WIU & MPV	71.3	83.6	84.7	97.8	97.6	94.5	89.4	83.0	87.7 *(-2.9)*

Table 2: Ablation experiment on GPT-3.5-Turbo. "*w/o WIU*" refers to removing *Wrong Information Utilization* (WIU). "*w/o MPV*" refers to removing *Multi-Perspective Verification* (MPV). "*w/o WIU & MPV*" refers to removing both *Wrong Information Utilization* and *Multi-Perspective Verification*.



	EoT Acc.	EoT F1	PoT Acc.	PoT F1
Assertion	82.2	85.2	79.8	87.6
Process	86.3	89.7	81.2	89.0
Result	84.8	88.0	84.2	91.1
Voting	88.2	90.9	84.8	91.4

Figure 4: Performance comparison results from various verification perspectives. "Voting" represents the final judgment after voting from the three perspectives.

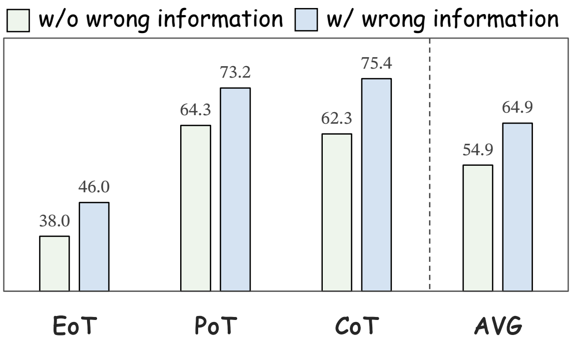


Figure 5: Comparison of performance without utilizing wrong reasoning information and with integrated wrong reasoning information.

Wrong Information Utilization exhibits a performance decrease across all datasets, with an average reduction of 1.0%. This demonstrates that incorporating wrong information can boost the reasoning performance of the LLMs, and more significant improvements can be achieved by utilizing more additional wrong reasoning information.

4.3.2 Answer 2: *Multi-Perspective Verification* can lead to more accurate judgments

To demonstrate that *Multi-Perspective Verification* can accurately judge the results generated by EoT and PoT, we directly evaluated the performance of the three perspectives and the final voting results of the three perspectives. For accurate assessment, we use accuracy (Acc.) and F1 score (F1) as evaluation metrics. Additionally, we evaluate the performance of the `WoT` framework without *Multi-Perspective Verification* to demonstrate the effectiveness of *Multi-Perspective Verification*.

The results are shown in Figure 4. We can directly observe that our proposed Process Verification and Result Verification outperform the Assertion Verification used in XoT with respect to accuracy and F1 score. Furthermore, the final Voting Verification further improves the accuracy. For EoT, Acc and F1 improved by 6% and 5.7%, respectively, while for PoT, they improved by 5% and 3.8%, respectively. Additionally, as shown in Table 2, the performance of `WoT` decreased by an average of 1.8% after the removal of *Multi-Perspective Verification*. This demonstrates the effectiveness of *Multi-Perspective Verification*, bringing significant benefits to overall performance improvement.

4.3.3 Answer 3: `WoT` can effectively minimize the reasoning steps needed

To compare the reasoning steps required by XoT and `WoT` in solving mathematical questions, we conduct experiments and record the average reasoning steps needed. As shown in Figure 6, the results indicate that `WoT` significantly reduces the

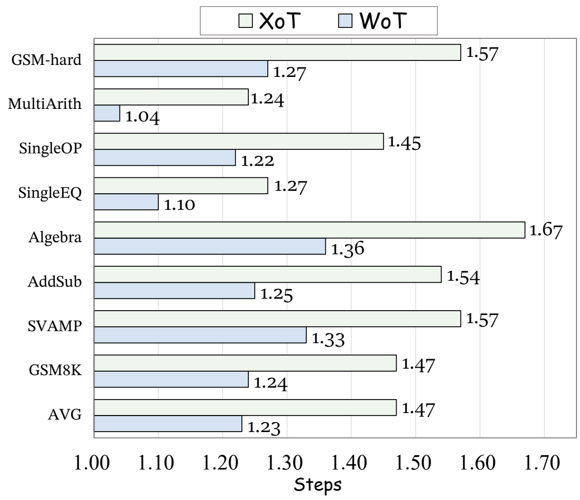


Figure 6: Comparison of the average reasoning steps required by XoT and `WoT` in solving questions.

Method	Acc.
CoT	42.2
PoT	70.3
EoT	53.4
XoT	71.3
WoT	**76.2**

Results on GSM-hard

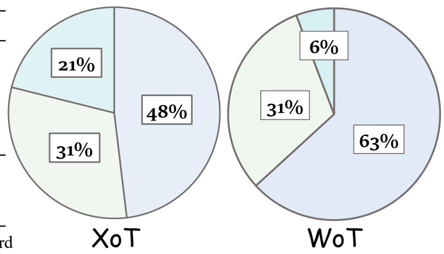


Figure 7: The proportion of reasoning methods ultimately used to solve questions by XoT and `WoT` on the GSM-hard dataset.

reasoning steps in each dataset, with an average reduction of 8% steps. This indirectly demonstrates the effectiveness of *Multi-Perspective Verification*, and *Wrong Information Utilization* in `WoT`. Accurate verification and efficient reasoning can effectively reduce the number of required reasoning steps, thereby enhancing reasoning efficiency.

4.3.4 Answer 4: Tips for solving difficult mathematical questions with `WoT`

To delve deeper into the reasons behind the significant performance improvement of `WoT` in solving reasoning challenges, we conduct a detailed analysis in this section. In the GSM-hard dataset, we extract the proportions of the methods ultimately used for reasoning, as shown in Figure 7. Our analysis reveals notable changes in the reasoning method proportions between XoT and `WoT`: the proportion of CoT decreased from 21% $\rightarrow$ 6%, while the proportion of PoT increased from 48% $\rightarrow$ 63%.

This change reflects the advantage of `WoT` in reasoning strategies. The numerical values in the GSM-hard dataset are usually large, often involving more than 10 digits. Because CoT reasoning has lower accuracy when handling large number value calculations, with an accuracy rate of only 42.2%. Since XoT relies more on CoT for reasoning, it results in lower accuracy. In contrast, `WoT` introduces a multiple perspectives verification mechanism, enabling more accurate judgment of reasoning results. Consequently, `WoT` more frequently adopts PoT for reasoning, thereby avoiding errors associated with CoT, and achieving significant overall improvement.

4.3.5 Answer 5: Qualitative analysis

To better comprehend `WoT`, we introduce a real-world example for qualitative analysis. As illustrated in Figure 8 (a), upon receiving a question, XoT selects EoT for reasoning. However, due to the limited reasoning capability of EoT, an incorrect result of "8" was generated. During Assertion Verification, this incorrect result was mistakenly identified as correct. As XoT relied solely on Assertion Verification, it erroneously output "8" as the final result. This example clearly illustrates the limitations of the single verification method and its adverse impact on reasoning accuracy.

In contrast, as shown in Figure 8 (b), `WoT`, when presented with the same question, initially also arrives at the incorrect answer "8". However, both Process Verification and Result Verification identified "8" as incorrect. Consequently, the system switches to PoT for the next reasoning step. In PoT reasoning, after being warned with a wrong example, PoT generates the correct reasoning and arrives at the correct result, "2". This result then passed verification from all three perspectives, ultimately confirming the correct answer, "2". This case further demonstrates the effectiveness of `WoT`, as combining three verification perspectives and utilizing wrong reasoning information significantly enhances reasoning capability.

5 Related Work

The rapid advancement of LLMs in recent years has introduced new opportunities in natural language processing (OpenAI, 2022; Team et al., 2023; Qin et al., 2024a,b). Particularly, the introduction of Chain-of-Thought (CoT) (Wei et al., 2022) opens a novel direction in this domain, attracting many researchers (Zhang et al., 2022; Fei et al., 2023, 2024; Zhang et al., 2024b; Xu et al., 2024a; Chen et al., 2024a). Specifically, Wei et al. (2022) propose using manually constructed CoT demonstra-

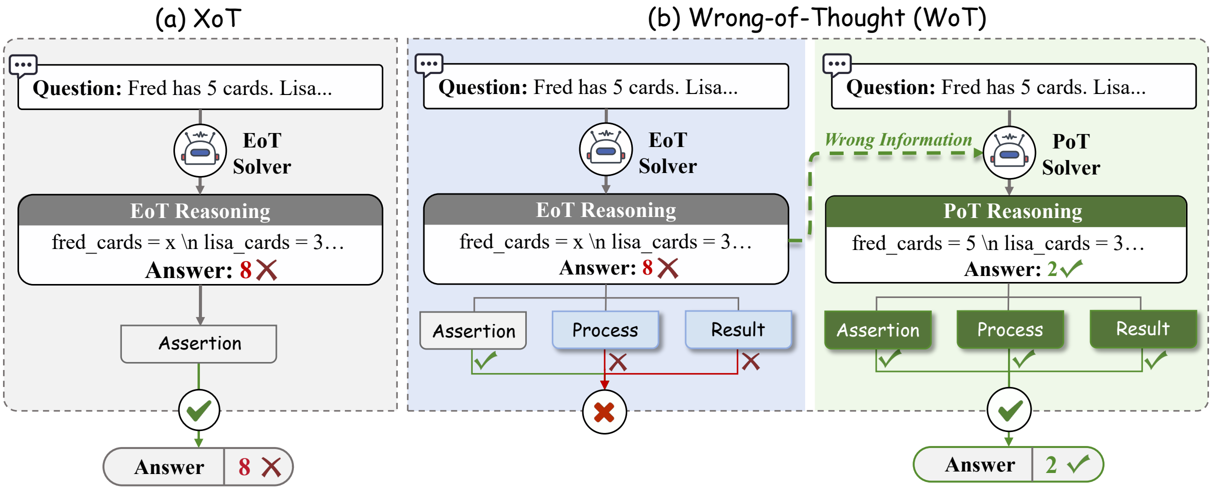


Figure 8: The case study. Figure (a) shows XoT reasoning, where it initially outputs an incorrect answer, "8". Assertion Verification mistakenly validated this as correct, resulting in the final wrong output of "8". Figure (b) shows WoT reasoning. EoT first outputs an incorrect answer, which was identified as wrong by Process and Result Verification, switching to PoT. Using the wrong reasoning of EoT, PoT arrived at the correct answer, "2". All three verification methods then confirmed this result, leading to the correct output of "2".

tions to enhance LLMs performance. Additionally, Chen et al. (2022) introduce the Program-of-Thoughts (PoT), enabling LLMs to generate Python programs to solve mathematical problems. Liu et al. (2023b) propose Equation-of-Thoughts (EoT), allowing LLMs to generate mathematical equations and then use external calculators to compute the results, offering a new perspective on problem-solving with LLMs. Chen et al. (2024b) propose a framework that iteratively explores and self-evaluates trees of thoughts, allowing LLMs to learn from trial and error and improve the quality of final answers. Xu et al. (2024b) propose transitioning LLMs from passive to active learning, thus enhancing their problem-solving capabilities. Zhou et al. (2024) present a method for LLMs to improve self-criticism and self-discovery, thereby forming explicit structures to enhance reasoning performance. Chen et al. (2023) propose using error code to implement code self-debug and improve the code generation capability of LLMs.

In the realm of nonlinear problem solving, Yao et al. (2023) introduce the Tree-of-Thoughts (ToT) framework, enabling LLMs to generate multiple reasoning paths to tackle mathematical reasoning tasks. Sel et al. (2023) propose the Algorithm-of-Thoughts (AoT), which not only generates multiple paths but also selects the optimal nodes, allowing for the repeated utilization of reasoning pathways. Besta et al. (2024) introduce Graph-of-Thoughts (GoT), a framework that models the information generated by LLMs as arbitrary graphs, enabling the synergistic integration of all reasoning processes. Ning et al. (2024) propose Skeleton-of-Thought (SoT), which first generates the skeleton of the answer and then utilizes LLMs for batched resolution, enhancing inference efficiency. Liu et al. (2023b) propose XoT, which integrates multiple reasoning thoughts and utilizes single assertion verification to decide whether to switch reasoning methods, achieving impressive results.

Compared to previous research, WoT employs multiple perspectives of verification while incorporating wrong information utilization. This greatly and effectively enhances overall reasoning performance. To our knowledge, this work is the first to incorporate *Multi-Perspective Verification* and *Wrong Information Utilization* within the continuously verifying and iterative framework.

6 Conclusion

In this work, we propose WoT, a framework that optimizes outputs by utilizing wrong information and multi-perspective verification. WoT comprises two core modules: *Multi-Perspective Verification* and *Wrong Information Utilization*. WoT achieves more accurate reasoning thought switching and utilizes wrong reasoning information. Extensive evaluations on eight datasets and five models demonstrate that WoT achieves superior performance. Furthermore, WoT exhibits powerful capabilities in difficult computation tasks.

Limitations

This work proposes a WoT framework to enhance verifying iteratively generated reasoning answers by *Multi-Perspective Verification* and *Wrong Information Utilization*. However, in our work, since "*Assertion Verification*" requires reliance on external rule executors, how to verify natural language-based CoT through assertions remains a question worthy of future research. Secondly, our verification method primarily validates the logical correctness of the model. Verifying the clarity and quality of the logical expression might further enhance the effectiveness of model reasoning. Finally, WoT may spend more tokens due to the incorporation of three verification perspectives and wrong reasoning information. We hope future work develops more efficient methods to address this challenge.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) via grant 62306342. This work was also sponsored by the Excellent Young Scientists Fund in Hunan Province (2024JJ4070) and the Science and Technology Innovation Program of Hunan Province under Grant 2024RC3024. This work was supported by the Key Laboratory of Computing Power Network and Information Security, Ministry of Education under Grant No.2023ZD032. We are grateful for resources from the High Performance Computing Center of Central South University.

References

Josh Achiam, Steven Adler, and Sandhini et al. Agarwal. 2023. Gpt-4 technical report. abs/2303.08774.

Jinze Bai, Shuai Bai, and Yunfei Chu et al. 2023. Qwen technical report. *ArXiv preprint*, abs/2309.16609.

Maciej Besta, Nils Blach, and Ales et al. Kubicek. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690.

Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. 2024a. M^{3}CoT: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8199–8221.

Sijia Chen, Baochun Li, and Di Niu. 2024b. Boosting of thoughts: Trial-and-error problem solving with large language models. In *The Twelfth International Conference on Learning Representations*.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*.

Xinyun Chen, Maxwell Lin, Nathanael Schaerli, and Denny Zhou. 2023. Teaching large language models to self-debug. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *ArXiv preprint*, abs/2110.14168.

Hao Fei, Bobo Li, Qian Liu, Lidong Bing, Fei Li, and Tat-Seng Chua. 2023. Reasoning implicit sentiment with chain-of-thought prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Short Papers)*.

Hao Fei, Shengqiong Wu, Wei Ji, Hanwang Zhang, Meishan Zhang, Mong-Li Lee, and Wynne Hsu. 2024. Video-of-thought: Step-by-step video reasoning from perception to cognition. In *Forty-first International Conference on Machine Learning*.

Luyu Gao, Aman Madaan, and Shuyan et al. Zhou. 2023. Pal: Program-aided language models. In *International Conference on Machine Learning*, pages 10764–10799. PMLR.

Joy He-Yueya, Gabriel Poesia, Rose E Wang, and Noah D Goodman. 2023. Solving math word problems by combining language models with symbolic solvers. *ArXiv preprint*, abs/2304.09102.

Mohammad Javad Hosseini, Hannaneh Hajishirzi, Oren Etzioni, and Nate Kushman. 2014. Learning to solve arithmetic word problems with verb categorization. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 523–533.

Albert Q Jiang, Alexandre Sablayrolles, and Arthur et al. Mensch. 2023. Mistral 7b. *ArXiv preprint*, abs/2310.06825.

Rik Koncel-Kedziorski, Hannaneh Hajishirzi, Ashish Sabharwal, Oren Etzioni, and Siena Dumas Ang. 2015. Parsing algebraic word problems into equations. *Transactions of the Association for Computational Linguistics*, 3:585–597.

Hanmeng Liu, Zhiyang Teng, Leyang Cui, Chaoli Zhang, Qiji Zhou, and Yue Zhang. 2023a. Logicot: Logical chain-of-thought instruction tuning. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Tengxiao Liu, Qipeng Guo, Yuqing Yang, Xiangkun Hu, Yue Zhang, Xipeng Qiu, and Zheng Zhang. 2023b. Plan, verify and switch: Integrated reasoning with diverse X-of-thoughts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2807–2822.

Aman Madaan, Niket Tandon, and Prakhar et al. Gupta. 2023. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36.

Xuefei Ning, Zinan Lin, and Zixuan et al. Zhou. 2024. Skeleton-of-thought: Prompting llms for efficient parallel generation. In *The Twelfth International Conference on Learning Representations*.

OpenAI. 2022. Chatgpt.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094.

Jingyuan Qi, Zhiyang Xu, Ying Shen, and et al. 2023. The art of socratic questioning: Recursive thinking with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4177–4199.

Libo Qin, Qiguang Chen, Xiachong Feng, Yang Wu, Yongheng Zhang, Yinghui Li, Min Li, Wanxiang Che, and Philip S Yu. 2024a. Large language models meet nlp: A survey. *ArXiv preprint*, abs/2405.12819.

Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. 2023. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2695–2709.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S Yu. 2024b. Multilingual large language model: A survey of resources, taxonomy and frontiers. *ArXiv preprint*, abs/2404.04925.

Subhro Roy and Dan Roth. 2015. Solving general arithmetic word problems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1743–1752.

Subhro Roy, Tim Vieira, and Dan Roth. 2015. Reasoning about quantities in natural language. *Transactions of the Association for Computational Linguistics*, Volume 3:1–13.

Bilgehan Sel, Ahmad Tawaha, and Vanshaj et al. Khattar. 2023. Algorithm of thoughts: Enhancing exploration of ideas in large language models. In *Forty-first International Conference on Machine Learning*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *ArXiv preprint*, abs/2312.11805.

Yongqi Tong, Dawei Li, and Sizhe et al. Wang. 2024. Can llms learn from previous mistakes? investigating llms' errors to boost for reasoning. *ArXiv preprint*, abs/2403.20046.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288.

Jason Wei, Xuezhi Wang, and Dale et al. Schuurmans. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. 2024a. Faithful logical reasoning via symbolic chain-of-thought. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13326–13365.

Zhipeng Xu, Zhenghao Liu, Yibin Liu, and et al. 2024b. ActiveRAG: Revealing the treasures of knowledge via active learning. *ArXiv preprint*, abs/2402.13547.

Shunyu Yao, Dian Yu, and Jeffrey et al. Zhao. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Tianjun Zhang, Aman Madaan, Luyu Gao, Steven Zhang, Swaroop Mishra, Yiming Yang, Niket Tandon, and Uri Alon. 2024a. In-context principle learning from mistakes. In *ICML 2024 Workshop on In-Context Learning*.

Yongheng Zhang, Qiguang Chen, Min Li, Wanxiang Che, and Libo Qin. 2024b. AutoCAP: Towards automatic cross-lingual alignment planning for zero-shot chain-of-thought. In *Findings of the Association for Computational Linguistics 2024*, pages 9191–9200.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. 2024. Progressive-hint prompting improves reasoning in large language models. In *AI for Math Workshop @ ICML 2024*.

Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V Le, Ed H Chi, Denny Zhou, Swaroop Mishra, and Huaixiu Steven Zheng. 2024. Self-discover: Large language models self-compose reasoning structures. *arXiv preprint arXiv:2402.03620*.