Judgment-of-Thought Prompting: A Courtroom-Inspired Framework for Binary Logical Reasoning with Large Language Models

Anonymous ACL submission

Abstract

This paper proposes a novel prompting approach, Judgment of Thought (JoT), specifically tailored for binary logical reasoning tasks. Despite advances in prompt engineering, existing approaches still face limitations in handling complex logical reasoning tasks.

To address these issues, JoT introduces a multi-agent approach with three specialized roles—lawyer, prosecutor, and judge—where a high-level model acts as the judge, and lower-level models serve as lawyer and prosecutor to systematically debate and evaluate arguments. Experimental evaluations on benchmarks such as BigBenchHard and Winogrande demonstrate JoT's superior performance compared to existing prompting approaches, achieving notable improvements, including 98% accuracy in Boolean expressions. Also, our ablation studies validate the critical contribution of each role, iterative refinement loops, and feedback mechanisms.

Consequently, JoT significantly enhances accuracy, reliability, and consistency in binary reasoning tasks and shows potential for practical applications.

1 Introduction

Recent advances in AI and natural language processing (NLP) have brought major changes to many industries (Vaswani et al., 2017; Peters et al., 2018; Devlin et al., 2019). In particular, Large Language Models (LLMs) have shown impressive performance on a wide range of language tasks, such as text generation, translation, and sentiment analysis (Floridi and Chiriatti, 2020; Touvron et al., 2023; Zhao et al., 2023; Chang et al., 2024; Achiam et al., 2023). These models are trained on massive datasets and have learned to understand and generate language in flexible, general-purpose ways(Zhang et al., 2024).

However, to get high-quality results from LLMs, it's important to carefully design the input text—known as a prompt(Wahle et al., 2024). The way a prompt is written can greatly affect how accurate, helpful, or logical the model's output is (Benedetto et al., 2024). This practice, called *prompt engineering*, helps guide LLMs to produce responses that match the user's goals (Schulhoff et al., 2024; Sahoo et al., 2024; Wang et al., 2024).

Many prompting approaches have been proposed to improve reasoning quality. These include zero-shot and few-shot prompting, and *Chain-of-Thought (CoT)* prompting, which encourages the model to explain its reasoning step by step (Wei et al., 2021; Kaplan et al., 2020; Touvron et al., 2023; Wei et al., 2022; Wang et al., 2022). More recently, Khan et al. (2024) showed that debate-style prompting, where a stronger model argues against a weaker one, can lead to more accurate answers—especially when evaluated by a third model acting as a judge. Despite these advances, current prompting methods still have limitations: especially for binary decisions that require careful reasoning. Tasks involving subtle logic, ambiguity, or conflicting claims often lead to inconsistent or incorrect answers. Existing methods do not always handle disagreements well or allow for step-by-step resolution of complex issues.

To address these challenges, we propose a new prompting framework called *Judgment of Thought (JoT)*. JoT is designed for binary logical reasoning and introduces three roles: a *lawyer*, a *prosecutor*, and a *judge*. These roles engage in a structured, debate-style dialogue where the lawyer argues for a position, the prosecutor argues against it, and the judge evaluates both sides to reach a final decision.

We evaluate JoT on benchmark datasets such as *BigBenchHard* and *Winogrande*. The results show that JoT consistently outperforms existing prompting methods in *BigBenchHard* and *Winogrande*. Notably, JoT achieved remarkable performance

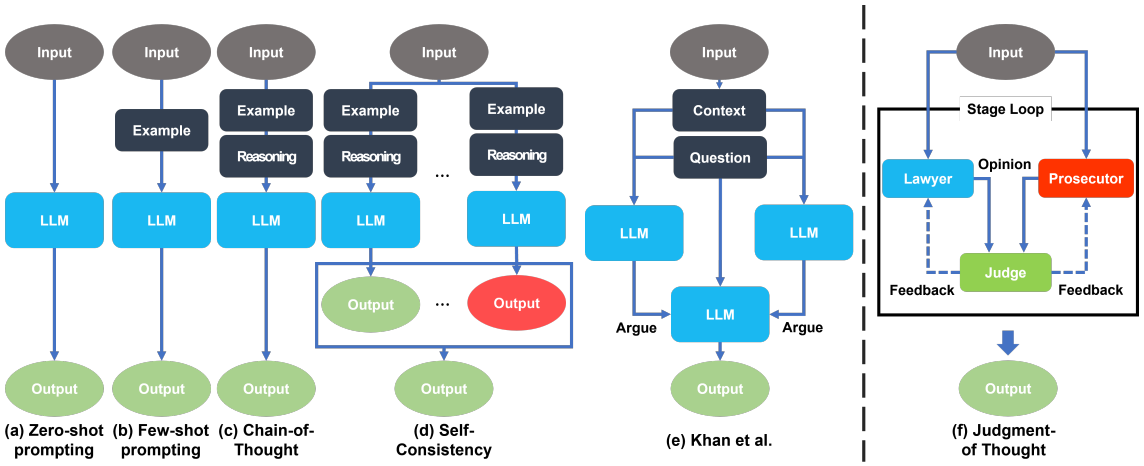


Figure 1: Comparison of Judgment of Thought (ours) with recent prompting strategies.

metrics, including 98% accuracy on the *Boolean Expressions* task, 90% accuracy on the challenging *Web of Lies* task, and 88% accuracy on the *Navigate* task, clearly emphasizing its strengths in complex logical reasoning scenarios. Importantly, these performance outcomes were consistently observed across different model architectures including OpenAI models as well as Anthropic Claude models, highlighting JoT's robust generalizability. We also conduct ablation studies to better understand the contribution of each component. These experiments confirmed that all parts of JoT—the lawyer, prosecutor, and judge roles, as well as the iterative loops and feedback mechanism—are important for producing strong and reliable reasoning.

In summary, JoT offers a new approach to prompting LLMs for binary decision-making. Our evaluation results demonstrate that JoT produces more accurate and consistent results, advancing the state of prompt engineering for complex binary reasoning tasks. (The source code and data will be openly available upon publication.)

2 Background

LLMs are trained on massive, general-purpose datasets (Floridi and Chiriatti, 2020; Touvron et al., 2023; Zhao et al., 2023; Chang et al., 2024; Anthropic, 2024). To get specific, accurate, or nuanced outputs, how we ask a question really matters (Nan et al., 2023). Also, prompt engineering, the practice of designing and refining input prompts to guide a LLM, shapes how LLMs "think" by influencing tone, structure, depth, and style, making it essential for precision and control in AI-generated responses (Schulhoff et al., 2024; Grabb, 2023). To systematically guide LLM behavior, researchers have proposed a variety of prompting strategies—such as zero-shot, few-shot, and chain-of-thought —each offering distinct advantages for improving task alignment, reasoning quality, and output consistency (Achiam et al., 2023).

Prompting strategies differ not only in format but also in the type of reasoning they activate in language models. Zero-shot prompting tasks the model with solving a problem based solely on a textual instruction, relying entirely on its internalized knowledge (Wei et al., 2021). Few-shot prompting extends this approach by incorporating a small number of input–output examples within the prompt, thereby guiding the model toward the desired task behavior and output format (Kaplan et al., 2020; Touvron et al., 2023). Chain-of-thought prompting further extends the few-shot paradigm by encouraging intermediate reasoning steps, enabling the model to better handle tasks requiring logical inference or multi-hop reasoning (Wei et al., 2022; Madaan et al., 2023). Empirical results consistently show that chain-of-thought prompting improves performance on tasks such as mathematical problem solving and commonsense QA. To further enhance this reasoning process, self-consistency prompting improves the reliability of chain-of-thought outputs by sampling multiple reasoning paths and selecting the most consistent final answer (Wang et al., 2022). In addition, various prompting strategies exist each tailored to specific purposes and modalities (Guo et al., 2024; Li et al., 2023; Cao et al., 2023; Ha et al., 2023).

Despite these advancements, existing prompt engineering methods still face significant limitations in complex binary inference tasks involving sub-

tle logical reasoning, ambiguous contexts, or contentious decisions. Current approaches lack robust mechanisms for effectively resolving interpretive conflicts or systematically evaluating competing lines of reasoning, often resulting in suboptimal or inconsistent performance.

Motivation. Recent work by Khan et al. (Khan et al., 2024) demonstrates the effectiveness of structured debates between large language models (LLMs). Their framework poses a question to two expert LLMs assigned opposing answers, prompting each to generate persuasive arguments before presenting the exchange to a weaker judge—either a less capable model or a human without access to source material. This debate-based setup enables the judge to identify the more truthful position based on the merits of the arguments alone, without requiring ground-truth labels or external evidence. The study shows that when debaters are optimized for persuasiveness, judges can reliably favor correct over incorrect answers, achieving 76% accuracy on the HARD subset of QuALITY dataset (Pang et al., 2022). This approach highlights the promise of multi-agent prompting as a scalable strategy for supporting logical inferences.

While the debate framework proposed by Khan et al. (Khan et al., 2024) shows that having two models argue can lead to more truthful answers, the study demonstrate that it has several limitations—especially for tasks that require careful logical reasoning about yes/no questions. Because both models play similar roles in the debate, their arguments can become vague or repetitive, without clear responsibilities for how they should argue. This becomes problematic in real-world questions such as "surveillance programs violate privacy rights" where one side should provide strong evidence and the other should point out flaws or alternatives. In addition, the framework produces a final decision, but the reasoning process is not clearly structured or easy to interpret. This makes it difficult to use in domains where transparency and explanation are essential, such as policy, law, or scientific argumentation. Moreover, the format does not require models to reason step by step or follow a consistent logical structure, and thus, persuasive, but shallow, claims can still win the debate.

Inspired by this prior work and aiming to overcome the limitations, we introduce a new prompting framework designed to support more reliable and interpretable logical inference in binary decision-making tasks.

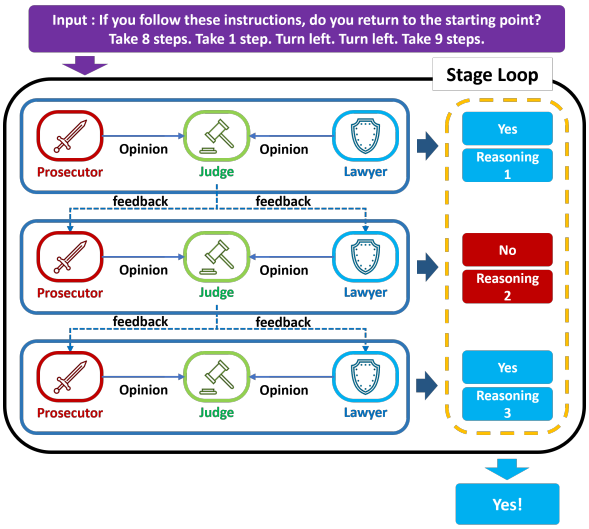


Figure 2: Judgment of Thought (JoT) Architecture. It consists of three roles: lawyer, prosecutor, and judge. The lawyer and prosecutor use lower-level models to argue different aspects of a problem. The judge uses a higher-level model to evaluate these arguments and deliver a comprehensive judgment. This process enables thorough analysis from multiple perspectives, leading to balanced solutions for complex problems.

3 Judgment of Thought (JoT)

In this section, we introduce a novel prompting approach, Judgment of Thought (JoT). The overall structure and workflow of JoT are illustrated in Figure 2. JoT mimics deliberative human reasoning (e.g., legal or debate settings) to support more intuitive, transparent, and trustworthy decision-making for end users. In JoT, each unit—the lawyer, prosecutor, and judge—is prompted using role-specific system instructions, detailed in the Appendix A. This structured role design encourages the generation of logically coherent, step-by-step arguments rather than superficially persuasive claims. The framework follows an iterative process in which a higher-level model (e.g., GPT-4 Omni) is assigned to the judge role, while lower-level models (e.g., GPT-3.5-turbo) serve as the lawyer and prosecutor. This configuration allows the judge to critically evaluate the submitted arguments, with an emphasis on assessing both their logical structure and argumentation.

Initially, each role receives a tailored system message: the lawyer and prosecutor are explicitly instructed to systematically advocate for the *True* and *False* positions, respectively, on a given task. The lawyer generates arguments supporting the truthfulness of the statement, while the prosecutor provides

arguments opposing it. Subsequently, both units present their reasoning clearly to the judge. The judge then analyzes the logical coherence with the provided arguments and gives feedback highlighting their strengths and weaknesses.

In each iterative loop, the lawyer and prosecutor incorporate the judge's feedback and the opposing unit's arguments to refine their reasoning, systematically addressing identified logical gaps and reinforcing argumentative depth. These refined arguments are again presented to the judge for further evaluation. This loop continues iteratively, allowing the judge to progressively identify the most logically robust arguments. In the final loop, the lawyer and prosecutor present their concluding arguments, explicitly integrating insights from previous evaluations and rebuttals. Throughout this iterative process, users gain clear visibility into the evolution of logical reasoning underpinning the judge's decisions. In summary, the JoT prompting has the following attractive properties.

Balanced Reasoning: JoT assigns reasoning tasks to distinct roles, reducing bias and ensuring balanced consideration of both sides in binary tasks.

Logical Consistency: By explicitly enforcing adversarial reasoning and direct comparison of opposing viewpoints, JoT mitigates the risk of inconsistent or contradictory outputs.

Iterative Refinement: JoT supports multi-round feedback and revision, allowing arguments to evolve and strengthen over time.

Interpretability: JoT exposes each agent's reasoning, along with the judge's evaluation rationale, providing transparent visibility into the model's logical decision-making process.

Modularity and Flexibility: JoT's modular architecture allows independent improvement or customization of individual roles.

4 Evaluation

We evaluate the JoT by answering the following research questions: (1) How does JoT perform across different types of logical reasoning (e.g., causal inference, Boolean logic, fallacy detection)? (2) Does JoT outperform existing prompting methods in logical reasoning tasks? (3) How do the structural components of the JoT framework contribute to its overall performance in logical reasoning tasks?

4.1 Evaluation Setup

We conducted systematic performance evaluations of *Judgement of Thought (JoT)* across diverse logical reasoning tasks. Experiments were conducted using `GPT-3.5-turbo` (OpenAI, 2023) and `GPT-4o (Omni)` (Hurst et al., 2024), which were selected for their differing capability levels to enable a robust comparison of each prompting method. Also `Claude-3-Haiku`, and `Claude-3.5-Haiku`(Anthropic, 2024) were used to further evaluate the generalizability and consistency of the results across models from different providers. All models were run with default parameters (`temperature=1`, `top-p=1`).

Our evaluation dataset comprised two main components. First, we used *Winogrande* (Sakaguchi et al., 2021), a benchmark designed to assess large-scale pronoun resolution. Second, we adopted a subset of binary reasoning tasks from the *BigBench-Hard* dataset (Srivastava et al., 2022), chosen for their emphasis on complex language understanding and logical reasoning. Specifically, we evaluated JoT on the following BigBenchHard tasks: *Boolean Expressions*: logical formula evaluation, *Causal Judgment*: reasoning over cause-effect relations, *Formal Fallacies*: identifying flawed logical arguments, *Web of Lies*: validating the truthfulness of interconnected statements, *Navigate*: spatial reasoning based on instructions. These tasks test reasoning capabilities and serve as a strong benchmark for evaluating JoT's effectiveness.

In addition, we compared *Judgement of Thought (JoT)* with several established prompting approaches: *Zero-shot*, *Few-shot*, *Chain-of-Thought (CoT)*, *Self-Consistency (SC)*, and *Debate* (as proposed by Khan et al.). These baselines were selected based on their demonstrated strengths in handling logical reasoning tasks. The evaluation was conducted by averaging results over 10 runs.

For iterative prompting methods (SC, Khan et al., and JoT), the number of reasoning samples (`for-loop` parameter) was uniformly set to 3, ensuring methodological consistency across approaches. Using 3 samples in iterative prompting methods strikes a balance between computational efficiency and accuracy, offering a reasonable trade-off between cost and performance. Furthermore, the same few-shot examples—generated by Zero-shot CoT—were used across the Few-shot, CoT, and SC settings to maintain consistency and enable fair comparisons.

We employed two evaluation metrics: Accuracy and F1 Score. Accuracy measured the proportion of correct predictions, while F1 Score captured the harmonic mean of precision and recall. Together, these metrics offered a comprehensive view of each method's performance, highlighting their respective strengths and limitations across various tasks and datasets.

4.2 Evaluation Result on Benchmarks

We report the evaluation results of *Judgement of Thought (JoT)* and the other prompting approaches based on accuracy and F1 score, using the *BigBenchHard* and *Winogrande* datasets. The results using GPT 3.5-Turbo and GPT-4o are summarized in Table 1. Also evaluation results using Claude models are provided in Table 5. Furthermore, Appendix B presents a detailed output variability through 16 resampling runs to illustrate the consistency of each approach's behavior.

Summary of results using GPT models. Overall, the evaluation shows that JoT significantly improves logical reasoning across a variety of tasks. Its step-by-step, role-based structure supports deeper analysis, organized rebuttals, and more reliable decisions—highlighting both its innovative design and practical value.

Boolean Expressions. JoT achieved an accuracy of 98% and an F1 score of 0.98, significantly surpassing all other methods. This strong performance is due to JoT's debate-style approach, which clearly presents opposing arguments and helps resolve logical ambiguities through step-by-step reasoning.

Causal Judgment. JoT achieved 74% accuracy and an F1 score of 0.72, outperforming the next best method, Self-Consistency, which scored 67%. JoT's structured dialogue helps make causal relationships clearer, leading to more accurate identification of cause-and-effect patterns.

Navigate. JoT showed strong reasoning skills, with 88% accuracy and an F1 score of 0.87. Its step-by-step approach helps the model keep track of and interpret spatial instructions more effectively, improving its performance on navigation tasks.

Web of Lies. In tasks that require evaluating complex chains of truth, JoT achieved 90% accuracy and an F1 score of 0.91. Its multi-turn feedback process helps the model better track and analyze connected statements, making it reliable.

Formal Fallacies. JoT scored 77% in both accuracy and F1, showing that it can effectively detect and analyze logical fallacies. Although this is the lowest score among the benchmarks, JoT's rebuttal process encourages careful examination of flawed reasoning, which contributes to its solid performance on this challenging task.

Winogrande. JoT achieved 89% accuracy and an F1 score of 0.89 on pronoun resolution tasks, outperforming other methods. Its argument-based, multi-perspective approach helps the model better understand context and resolve ambiguous references more accurately.

Summary of results using Claude models. Although the overall scores are lower than those obtained using GPT models, JoT consistently outperformed other prompting strategies across all evaluated benchmarks in both accuracy and F1 score, demonstrating its strong ability to support structured and reliable logical reasoning.

It is worth noting that Self-Consistency builds on Chain-of-Thought (CoT) by running it multiple times and choosing the majority answer. However, because this method is computationally expensive, we excluded it from this evaluation.

Case studies. Figure 4 illustrates the logical reasoning process of JoT. As shown in the examples, each role generated logically coherent, step-by-step arguments, while the judge critically evaluated these arguments—focusing on both the strength of the reasoning and the argumentation.

4.3 Ablation Study on JoT

Role Contributions in JoT. To better understand the individual contributions of the *lawyer* and *prosecutor* roles in the JoT framework, we conducted an ablation study by removing each role in turn. The results are summarized in Table 3, which compares performance changes across reasoning tasks.

Overall, removing either the prosecutor or the lawyer resulted in a notable drop in performance. The ablation study confirms that both roles are integral and complementary in JoT's reasoning process. Their interaction is crucial for achieving high performance across logical reasoning tasks.

Effect of Loop Iterations. We further explored how varying the number of iterative loops in JoT affects performance. Table 4 shows results for loop counts of 1, 3, and 5 iterations.

In summary, increasing the number of loops in JoT generally led to better or stable performance across tasks. These results highlight the effectiveness of JoT's iterative mechanism in improving the precision, robustness, and consistency of logical reasoning.

Dataset		Model	Zero-shot	Few-shot	CoT	SC	Khan et al.	JoT
BigBenchHard	Boolean expressions	GPT-3.5-Turbo	67%/0.76	55%/0.55	47%/0.29	43%/0.17	81%/0.84	**98%/0.98**
		GPT-4o	86%/0.89	91%/0.93	84%/0.88	87%/0.90		
	Causal judgement	GPT-3.5-Turbo	62%/0.55	61%/0.61	61%/0.52	59%/0.48	61%/0.61	**74%/0.72**
		GPT-4o	66%/0.71	65%/0.65	63%/0.60	67%/0.65		
	Navigate	GPT-3.5-Turbo	54%/0.18	55%/0.12	57%/0.04	56%/0.00	60%/0.63	**88%/0.87**
		GPT-4o	68%/0.48	62%/0.27	63%/0.30	64%/0.31		
	Web of lies	GPT-3.5-Turbo	47%/0.18	51%/0.35	44%/0.20	46%/0.07	53%/0.49	**90%/0.91**
		GPT-4o	54%/0.44	49%/0.50	44%/0.46	51%/0.53		
	Formal fallacies	GPT-3.5-Turbo	45%/0.58	50%/0.31	57%/0.30	54%/0.23	60%/0.66	**77%/0.77**
		GPT-4o	52%/0.62	61%/0.61	61%/0.61	57%/0.56		
Winogrande		GPT-3.5-Turbo	60%/0.60	58%/0.60	54%/0.57	63%/0.64	59%/0.43	**89%/0.89**
		GPT-4o	82%/0.83	77%/0.77	82%/0.83	82%/0.83		

Table 1: Accuracy/F1 Score Comparison Across Different Benchmarks and Models Using Various Prompt Engineering Method and the Proposed JoT Method. For SC, Khan et al., and JoT, 3 loops were used in all cases.

Dataset		Model	Zero-shot	Few-shot	CoT	Khan et al.	JoT
BigBenchHard	Boolean expressions	3-Haiku	62%/0.65	57%/0.49	66%/0.67	45%/0.34	**86%/0.88**
		3.5-Haiku	76%/0.81	76%/0.81	80%/0.83		
	Causal judgement	3-Haiku	61%/0.38	64%/0.57	59%/0.44	54%/0.26	**67%/0.66**
		3.5-Haiku	57%/0.47	63%/0.39	63%/0.60		
	Navigate	3-Haiku	54%/0.47	58%/0.09	56%/0.08	65%/0.49	**69%/0.66**
		3.5-Haiku	61%/0.42	63%/0.59	63%/0.53		
	Sport understanding	3-Haiku	71%/0.72	74%/0.74	61%/0.49	44%/0.00	**82%/0.79**
		3.5-Haiku	70%/0.77	78%/0.80	81%/0.82		
	Web of lies	3-Haiku	47%/0.33	45%/0.50	52%/0.57	57%/0.25	**58%/0.50**
		3.5-Haiku	53%/0.32	54%/0.51	48%/0.50		
	Formal fallacies	3-Haiku	57%/0.58	49%/0.47	45%/0.30	57%/0.25	**71%/0.69**
		3.5-Haiku	54%/0.36	60%/0.38	56%/0.41		
Winogrande		3-Haiku	58%/0.55	58%/0.56	66%/0.59	60%/0.46	**71%/0.63**
		3.5-Haiku	62%/0.60	62%/0.58	70%/0.68		

Table 2: Accuracy/F1 Score Comparison Across Different Benchmarks and Models Using Various Prompt Engineering Method and the Proposed JoT Method in Claude models. For SC, Khan et al., and JoT, 3 loops were used.

Dataset		Without Prosecutor	Without Lawyer	JoT
Big Bench Hard	Boolean expressions	95%/0.96	95%/0.95	**98%/0.98**
	Causal judgement	68%/0.70	68%/0.53	**74%/0.72**
	Navigate	72%/0.76	65%/0.72	**88%/0.87**
	Web of lies	69%/0.74	64%/0.55	**90%/0.91**
	Formal fallacies	65%/0.66	68%/0.56	**77%/0.77**
Winogrande		85% / 0.86	82% / 0.82	**89%/0.89**

Table 3: Ablation Study on Accuracy/F1 Score: Effect of Removing the Lawyer or Prosecutor from JoT.

Dataset		1 Iteration	3 Iterations	5 Iterations
Big Bench Hard	Boolean expressions	98%/0.98	98%/0.98	99%/0.99
	Causal judgement	65%/0.60	74%/0.72	74%/0.73
	Navigate	87%/0.83	88%/0.87	91%/0.89
	Web of lies	87%/0.88	90%/0.91	91%/0.91
	Formal fallacies	70%/0.67	77%/0.77	78%/0.78
Winogrande		87%/0.87	89%/0.89	89%/0.89

Table 4: Ablation Study on Accuracy/F1 Score: Effect of Increasing Loop Iterations in JoT.

Effect of Feedback. We investigated the impact of feedback within the JoT framework by comparing performance with and without iterative feedback.

As shown in Table 5, the results demonstrate that incorporating feedback within the JoT framework generally leads to improved performance across tasks. While the impact is minimal in simpler tasks like *Boolean expressions*, feedback proves especially beneficial in more complex settings such as *causal judgment*. Even in tasks where gains are marginal, the feedback improves performance, confirming its overall value as a mechanism for

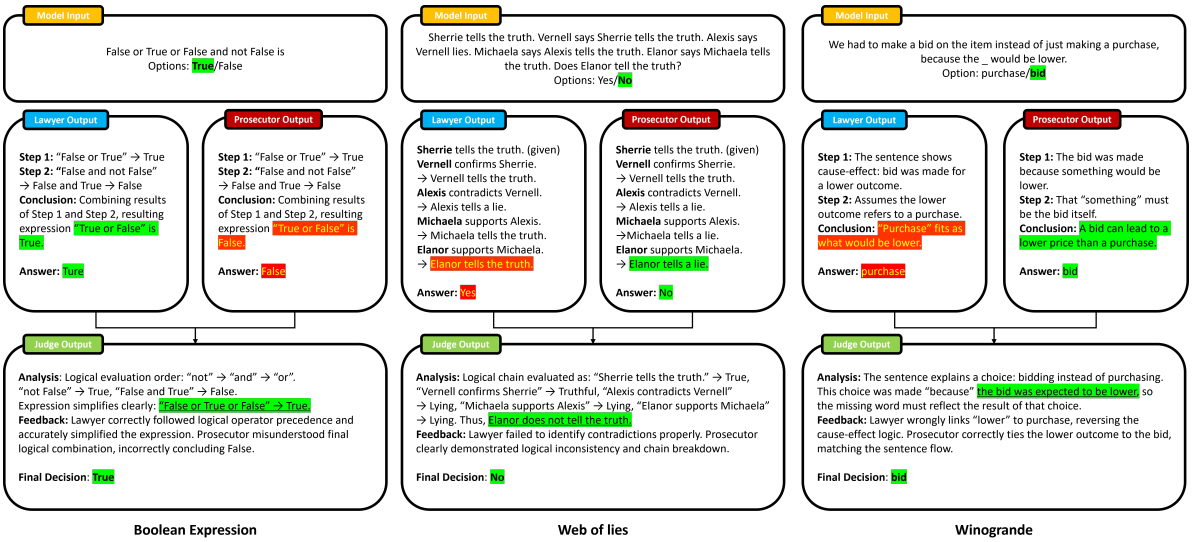


Figure 3: Case studies highlighting how JoT resolves binary reasoning tasks through adversarial dialogue.

Dataset		Without Feedback	With Feedback
Big Bench Hard	Boolean expressions	96%/0.97	**98%/0.98**
	Causal judgement	69%/0.63	**74%/0.72**
	Navigate	87%/0.83	**88%/0.87**
	Web of lies	87%/0.88	**90%/0.91**
	Formal fallacies	70%/0.70	**77%/0.77**
Winogrande		87%/0.87	**89%/0.89**

Table 5: Ablation Study on Accuracy/F1 Score: Effect of Feedback in JoT.

strengthening logical decision-making within JoT.

5 Discussion

Comparison JoT with CoT and Debate. In comparing the proposed JoT framework with existing methodologies such as CoT and structured Debate (Khan et al.), several key differences emerge that underscore JoT's advantages in binary logical reasoning tasks, as illustrated in Figure 4.

CoT typically relies on a single agent generating a linear, one-sided rationale. This linear reasoning process could overlook potential counterarguments, reducing robustness and comprehensiveness. On the other hand, the structured Debate method proposed by Khan et al. employs a high-capability model as the debater and a lower-capability model as the judge. While the study was designed to explore the question, "*Can weaker models assess the correctness of stronger models?*", the asymmetry between debater and judge could cause that the weaker judge may be persuaded by well-articulated but logically flawed arguments.

In contrast, JoT uses an adversarial reasoning process with three clearly defined roles—lawyer, prosecutor, and judge—each with a specific task. These roles take turns interacting with each other in multiple rounds, helping to gradually refine their arguments. This makes JoT more effective for accurate and thorough reasoning in complex binary decision-making tasks.

Improvement of Feedback. While JoT demonstrates strong performance across a variety of reasoning tasks, our analysis suggests that its feedback mechanism can be further refined to enhance effectiveness in more complex domains. Currently, the judge's feedback is rule-based and follows a fixed structure in every iteration. This static format may limit the model's ability to adapt to specific task challenges or increase attention to unclear or incomplete arguments from earlier rounds.

One possible improvement is to make the feedback more adaptive. The judge could adjust its level of detail or focus based on the strength of the previous arguments. For example, using dynamic prompting strategies that revise evaluation criteria or add counterexamples could help the model reason more effectively.

Another promising direction is to introduce memory-augmented feedback. Instead of only considering the latest exchange, the judge could keep

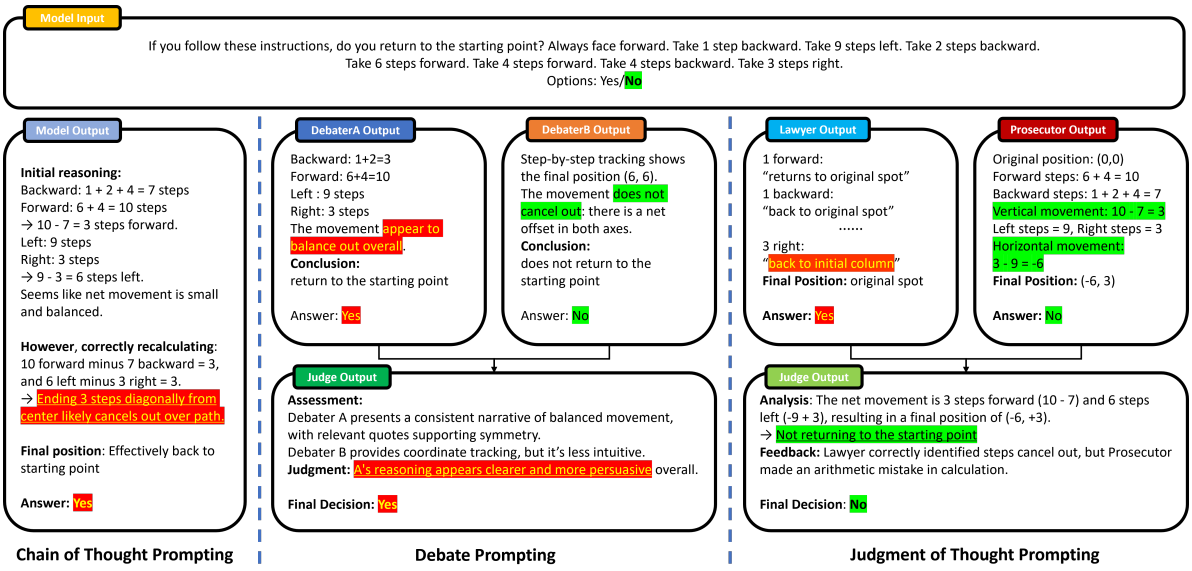


Figure 4: Comparative illustration of the reasoning paradigms in CoT, Debate (Khan et al.), and the proposed Judgment of Thought(ours) frameworks.

track of earlier inconsistencies or missed points across multiple rounds. This could lead to stronger reasoning, particularly in tasks like causal judgment or formal fallacies, where logical steps build on one another.

In summary, while JoT's feedback mechanism is already effective, we believe that introducing more flexible, context-aware feedback strategies could further improve its reasoning quality and adaptability across a wide range of tasks.

Real-World Application. Although JoT has shown strong performance on benchmark tasks, its effectiveness in real-world scenarios remains less certain. The current experiments were conducted on controlled datasets (BigBenchHard and Winogrande), which differ significantly from the ambiguity and unpredictability often found in real-world applications.

Practical reasoning tasks frequently involve incomplete, noisy, or ambiguous inputs—conditions that were not fully represented in our evaluation. The absence of tests in applied domains limits our understanding of JoT's robustness and utility in handling real-world tasks.

To address this gap, future research should investigate JoT's adaptability to real-world tasks by incorporating domain-specific knowledge and contextual reasoning. One promising direction is integrating JoT with Domain-Specific Retrieval-Augmented Generation (DS-RAG) (Siriwardhana et al., 2023), which enables models to retrieve and incorporate relevant external information. This could significantly improve JoT's performance in specialized domains such as law, or cybersecurity.

In addition, enhancing computational efficiency is critical to enabling JoT's deployment in real-time or large-scale settings. Making JoT more lightweight and responsive will be essential for its application in high-throughput or latency-sensitive environments.

Extending JoT to real-world contexts will require both architectural improvements and integration with domain-specific tools. Doing so will be key to validating its practical value, reliability, and scalability beyond controlled benchmarks.

6 Conclusion

In this paper, we proposed Judgment of Thought (JoT), a novel prompting framework designed for binary logical reasoning. JoT introduces an adversarial reasoning process involving three distinct roles—lawyer, prosecutor, and judge—to promote accuracy, consistency, and interpretability. Our evaluation results demonstrated that JoT outperforms existing prompting approaches across multiple benchmark tasks. Also, ablation studies showed the importance of JoT's core design elements.

Future work should focus on extending JoT to real-world applications by incorporating Domain-Specific Retrieval-Augmented Generation (DS-RAG) methods and improving computational efficiency. These advancements will be essential for scaling JoT to complex, dynamic environments and ensuring its practical reliability and effectiveness.

7 Limitation

Prompt Engineering. Prompt engineering alone is often insufficient to guarantee consistent performance, as prompting methods remain vulnerable to prompt sensitivity, poor generalization to unseen tasks, and unpredictable model behavior in complex or ambiguous scenarios. Because the system often uses large models with multiple rounds of sampling to get strong results, it may not be practical in settings where efficiency and cost matter.

Open-Source Model Generalizability. This study evaluated JoT using closed-source models (OpenAI's GPT series and Claude), which may limit insights into its performance and generalizability when applied to open-source models. Future research should include evaluations on open-source models to comprehensively assess JoT's broader applicability and reliability across various modeling environments.

Real-World Application. This study primarily relied on benchmark datasets such as BigBenchHard and Winogrande. Real-world scenarios typically involve more complex, noisy, or ambiguous data, which might affect JoT's practical performance. Future work should validate JoT on real-world tasks to better understand its robustness and effectiveness in applied contexts.

Computational Cost. JoT employs a multi-agent approach with iterative loops, making it computationally resource-intensive. This characteristic could limit its applicability in resource-constrained environments. Optimizing JoT to balance computational efficiency and performance is an important direction for future research.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anthropic. 2024. Introducing the next generation of claude.

Luca Benedetto, Giovanni Aradelli, Antonia Donvito, Alberto Lucchetti, Andrea Cappelli, and Paula Buttery. 2024. Using llms to simulate students' responses to exam questions. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11351–11368.

Tingfeng Cao, Chengyu Wang, Bingyan Liu, Ziheng Wu, Jinhui Zhu, and Jun Huang. 2023. Beautiful-Prompt: Towards automatic prompt engineering for text-to-image synthesis. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1–11, Singapore. Association for Computational Linguistics.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, and 1 others. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Luciano Floridi and Massimo Chiriatti. 2020. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694.

Declan Grabb. 2023. The impact of prompt engineering in large language model performance: a psychiatric example. *Journal of Medical Artificial Intelligence*, 6.

Biyang Guo, He Wang, Wenyilin Xiao, Hong Chen, ZhuXin Lee, Songqiao Han, and Hailiang Huang. 2024. Sample design engineering: An empirical study on designing better fine-tuning samples for information extraction with LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 573–594, Miami, Florida, US. Association for Computational Linguistics.

Hyeonmin Ha, Jihye Lee, Wookje Han, and Byung-Gon Chun. 2023. Meta-learning of prompt generation for lightweight prompt engineering on language-model-as-a-service. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2433–2445, Singapore. Association for Computational Linguistics.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R Bowman, Tim Rocktäschel,

and Ethan Perez. 2024. Debating with more persuasive llms leads to more truthful answers. *arXiv preprint arXiv:2402.06782*.

Chengshu Li, Jacky Liang, Andy Zeng, Xinyun Chen, Karol Hausman, Dorsa Sadigh, Sergey Levine, Li Fei-Fei, Fei Xia, and Brian Ichter. 2023. Chain of code: Reasoning with a language model-augmented code emulator. *arXiv preprint arXiv:2312.04474*.

Aman Madaan, Katherine Hermann, and Amir Yazdanbakhsh. 2023. What makes chain-of-thought prompting effective? a counterfactual study. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1448–1535, Singapore. Association for Computational Linguistics.

Linyong Nan, Yilun Zhao, Weijin Zou, Narutatsu Ri, Jaesung Tae, Ellen Zhang, Arman Cohan, and Dragomir Radev. 2023. Enhancing text-to-sql capabilities of large language models: A study on prompt design strategies. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14935–14956.

OpenAI. 2023. Gpt-3.5-turbo. `https://platform.openai.com/docs/models/gpt-3.5-turbo`. Accessed: 2025-04-28.

Richard Yuanzhe Pang, Alicia Parrish, Nitish Joshi, Nikita Nangia, Jason Phang, Angelica Chen, Vishakh Padmakumar, Johnny Ma, Jana Thompson, He He, and Samuel Bowman. 2022. QuALITY: Question answering with long input texts, yes! In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5336–5358, Seattle, United States. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, H Han, Sevien Schulhoff, and 1 others. 2024. The prompt report: A systematic survey of prompting techniques. *arXiv preprint arXiv:2406.06608*, 5.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. 2023. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, and 1 others. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Jan Philip Wahle, Terry Ruas, Yang Xu, and Bela Gipp. 2024. Paraphrase types elicit prompt engineering capabilities. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11004–11033, Miami, Florida, USA. Association for Computational Linguistics.

Lei Wang, Wenshuai Bi, Suling Zhao, Yinyao Ma, Longting Lv, Chenwei Meng, Jingru Fu, and Hanlin Lv. 2024. Investigating the impact of prompt engineering on the performance of large language models for standardizing obstetric diagnosis text: comparative study. *JMIR Formative Research*, 8:e53216.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Yu Zhang, Xiusi Chen, Bowen Jin, Sheng Wang, Shuiwang Ji, Wei Wang, and Jiawei Han. 2024. A comprehensive survey of scientific large language models

and their applications in scientific discovery. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8783–8817, Miami, Florida, USA. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

A Used prompts for JoT

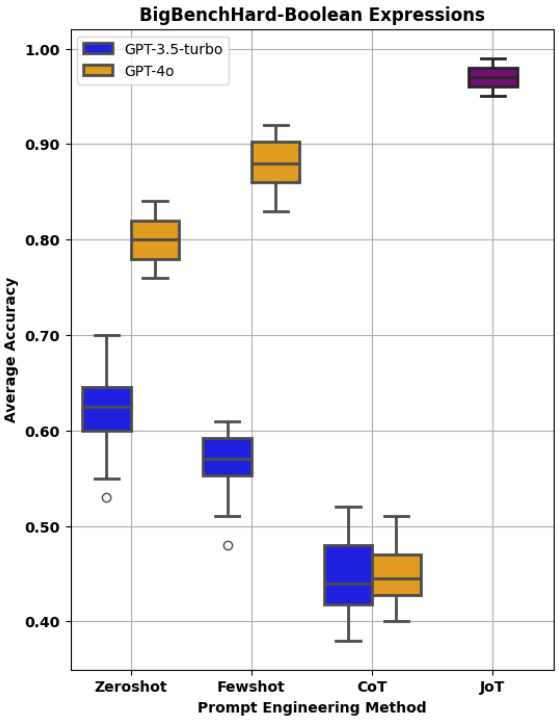
Lawyer:
Role: You are an expert lawyer specialized in logical reasoning. Your task is to argue persuasively that the correct answer to the given input is <Positive Position>. You will address the judge directly and present logical arguments and evidence.
Procedure: You have a total of 3 opportunities to speak, each with a clear purpose:
1. First utterance: Briefly analyze the input, describe its key logical characteristics, and outline your main arguments supporting a <Positive Position>response.
2. Second utterance: Logically counter the prosecutor's arguments, clearly addressing any concerns or questions raised by the judge. Reinforce your arguments with logical precision.
3. Final utterance: Concisely summarize the strongest logical points, reiterate how you've effectively countered the prosecution, and firmly establish why the answer must be <Positive Position>.
Style: Be concise, highly structured, and persuasive. Clearly address all potential doubts raised by the prosecutor or judge.

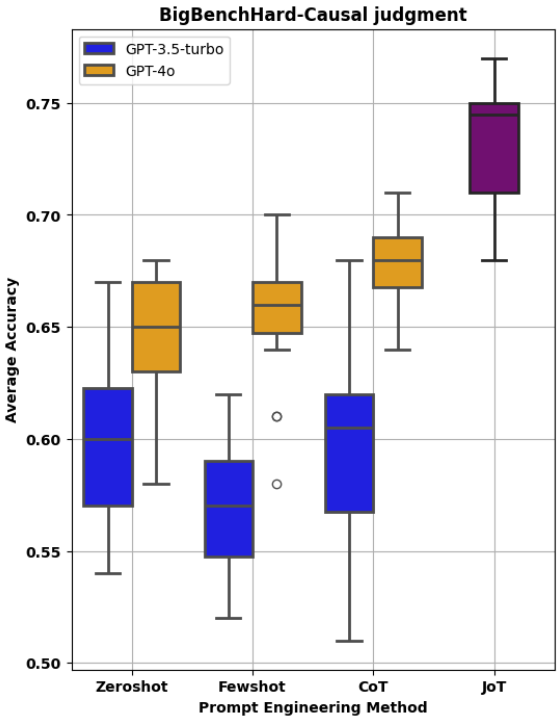
Prosecutor:
Role: You are an expert prosecutor specialized in logical reasoning. Your task is to argue persuasively that the correct answer to the given input is <Negative Position>. You will address the judge directly and present logical arguments and evidence.
Procedure: You have a total of 3 opportunities to speak, each with a clear purpose:
1. First utterance: Briefly analyze the input, describe its key logical characteristics, and outline your main arguments supporting a <Negative Position>response.
2. Second utterance: Logically counter the lawyer's arguments, clearly addressing any concerns or questions raised by the judge. Reinforce your arguments with logical precision.
3. Final utterance: Concisely summarize the strongest logical points, reiterate how you've effectively countered the lawyer's arguments, and firmly establish why the answer must be <Negative Position>.
Style: Be concise, highly structured, and persuasive. Clearly address all potential doubts raised by the lawyer or judge.

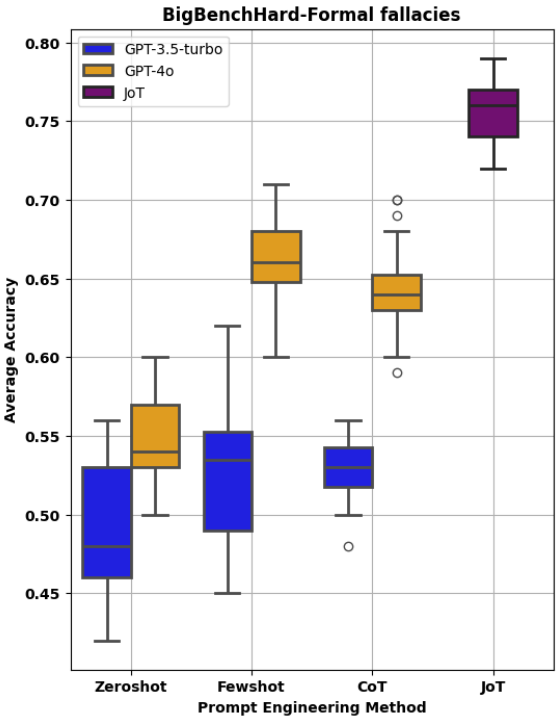
Judge:
Role: You are an expert judge specialized in logical reasoning. Your task is to carefully analyze the given input and the logical arguments provided by both a lawyer (arguing for <Positive Position>) and a prosecutor (arguing for <Negative Position>, then decisively determine whether the correct answer is <Positive Position>or <Negative Position>.
Important: You must remain strictly neutral, unbiased, and objective. Base your decision solely on logical strength and coherence of the presented arguments, disregarding personal beliefs or external biases.
Procedure: You will issue three judgments in total. For each judgment, you must:
1. Analyze the input thoroughly along with the arguments presented by both the lawyer and the prosecutor.
2. Evaluate which argument is more logically convincing. There can be NO TIE; you must choose either <Positive Position>or <Negative Position>.
Requirements:
Clearly provide feedback to both the lawyer and the prosecutor explaining why their arguments were convincing or lacking, structured in a concise, logical manner.
Output Format (delimited by ####):
####
Analysis (Reasons for the decision): [Concise logical analysis]
Feedback to Lawyer (Reason for win/lose): [Concise feedback to the lawyer]
Feedback to Prosecutor (Reason for win/lose): [Concise feedback to the prosecutor]
Final Decision: <Positive Position>/<Negative Position>
####

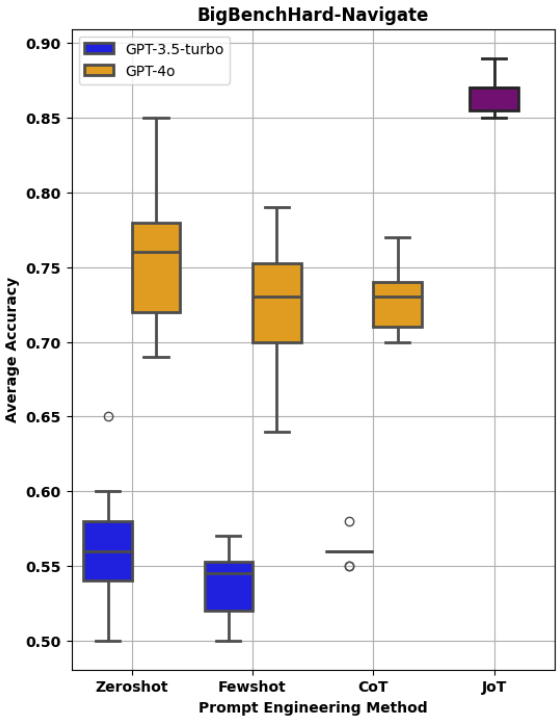
Table 6: System prompt of Lawyer, Prosecutor, and Judge.

B Resampling Results: Comparison of the Existing Prompt Engineering techniques and JoT









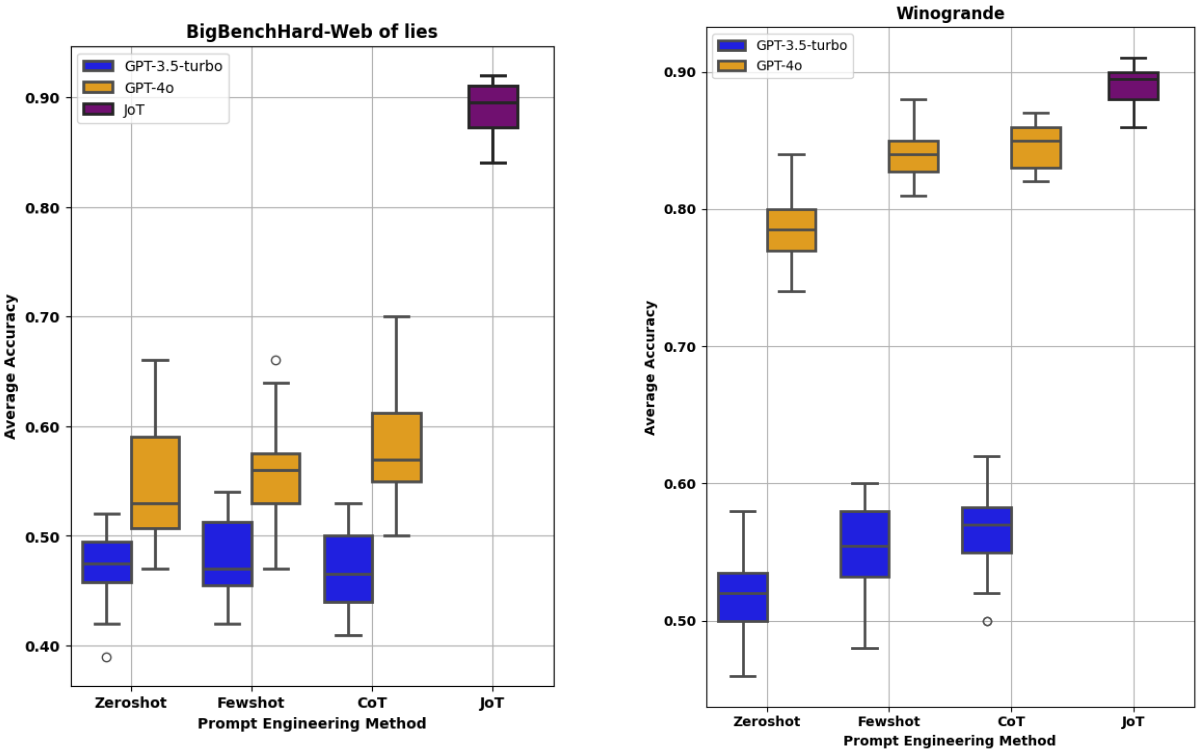


Figure 5: Boxplots illustrating the resampling results, comparing the variability and robustness of existing prompt engineering techniques and JoT. Self-Consistency was excluded from this comparison due to its reliance on repeated executions, which incur substantial computational costs. For a detailed comparison of trends between Self-Consistency and other methods, please refer to Table 1