

Inclusion-of-Thoughts: Mitigating Preference Instability via Purifying the Decision Space

Anonymous ACL submission

Abstract

Multiple-choice questions (MCQs) are widely used to evaluate large language models (LLMs). However, LLMs remain vulnerable to the presence of plausible distractors. This often diverts attention toward irrelevant choices, resulting in unstable oscillation between correct and incorrect answers. In this paper, we propose Inclusion-of-Thoughts (IoT), a progressive self-filtering strategy that is designed to mitigate this cognitive load (*i.e.*, instability of model preferences under the presence of distractors) and enable the model to focus more effectively on plausible answers. Our method operates to reconstruct the MCQ using only plausible option choices, providing a controlled setting for examining comparative judgements and therefore the stability of the model’s internal reasoning under perturbation. By explicitly documenting this filtering process, IoT also enhances the *transparency* and *interpretability* of the model’s decision-making. Extensive empirical evaluation demonstrates that IoT substantially boosts chain-of-thought performance across a range of arithmetic, commonsense reasoning, and educational benchmarks with *minimal* computational overhead.

1 Introduction

“The greatest gift we can offer each other is a second chance.” –Lloyd D. Newell

Although language models have demonstrated remarkable success across a range of NLP tasks, their ability to demonstrate reasoning is often seen as a limitation, which cannot be overcome solely by increasing model scale (Rae et al., 2022; bench authors, 2023). In an effort to address this shortcoming, Wei et al. (2023) have proposed chain-of-thought (CoT) prompting, where a language model is prompted to generate a series of short sentences that mimic the reasoning process a person might employ in solving a task. It has been observed that

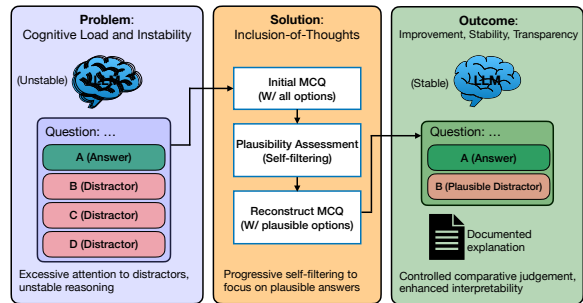


Figure 1: **Inclusion-of-Thoughts Framework** is a self-filtering strategy that reconstructs multiple-choice questions using only plausible options to mitigate model instability and enhance performance.

CoT prompting significantly improves model performance across a variety of multi-step reasoning tasks (Wei et al., 2023). However, failures on MCQ reasoning tasks are not solely caused by the absence of relevant knowledge, but also by instability in the model’s preference over options when exposed to multiple plausible distractors (Ma and Du, 2023; Fu et al., 2025; Balepur et al., 2024). That is, the correct answer already appears among the model’s high-confidence candidates, yet the final prediction fluctuates due to comparative ambiguity rather than missing reasoning steps (see Figure 3).

Rather than relying solely on training, recent research has emphasized the utility of increased test-time compute to elevate model performance (Welleck et al., 2024). One prominent approach involves sampling-based strategies that mimic test-time scaling; for instance, Wang et al. (2023) utilize majority voting over multiple generated samples to stabilize outputs. Alternatively, feedback-based techniques, such as Reflexion (Shinn et al., 2023a) and Self-Refine (Madaan et al., 2023), employ iterative loops where the model uses step-wise or outcome-based critiques to polish its responses. In this sense, Zhao et al. (2025) observe that applying simple sampling and self-verification allows

Gemini v1.5 Pro to exceed the performance benchmarks of o1-preview. Finally, structured search algorithms have also been adapted for reasoning; notably, Yao et al. (2023a) and Hao et al. (2023) implement global search strategies—ranging from BFS/DFS to Monte Carlo Tree Search—to systematically navigate the space of reasoning paths.

While these techniques have been proposed to improve model accuracy by leveraging additional compute during inference (Welleck et al., 2024), they often rely on expensive aggregation or complex search strategies. Hence, we introduce the **Inclusion-of-Thoughts (IoT)**, a novel approach designed to augment the standard CoT process by focusing on the stability of the model’s highest-ranked predictions in isolation. As illustrated in Figure 2, IoT achieves this by strategically *perturbing the input* to elicit and then isolate the model’s top two preferences, followed by a final, unconstrained comparative judgement. In contrast to other test-time methods, IoT is computationally lightweight and does not require extensive sampling, complex feedback loops, or tree search. It achieves its performance gain by resolving internal decision instability related to the presence of distractors, essentially purifying the decision space to focus on the model’s top-ranked hypotheses. This makes IoT not only a scalable and cost-effective method for improving reasoning stability beyond the gains achieved by simple aggregation, but also as a controlled test-time probe of preference stability under option-level perturbations. Another key advantage of IoT is its zero-shot, unsupervised implementation. It works entirely off-the-shelf with pre-trained language models, requiring no additional human annotation, auxiliary models, extra training, or fine-tuning.

We evaluate IoT on a wide range of arithmetic, commonsense reasoning, and educational tasks over 4 language models with varying scales: Llama-3.3-8B (AI@Meta, 2024), Olmo-2-7B/13B (OLMo et al., 2024), and GPT-4o-mini (OpenAI et al., 2024). Across all language models, IoT improves over CoT prompting by a striking margin across all tasks. In particular, when used with Olmo-2-7b, IoT achieves new state-of-the-art levels of performance across commonsense reasoning tasks, including OBQA (Mihaylov et al., 2018) (+3.40%), CSQA (Talmor et al., 2019) (+2.05%), GSM8K-MC (Cobbe et al., 2021) (+3.95%), and across educational benchmarks tasks such as ARC (Clark et al., 2018)

(+3.33%) and MMLU (Hendrycks et al., 2021) (+1.16%). Finally, the main contributions of our research are as follows:

1. We propose IoT, a lightweight, zero-shot test-time framework that improves MCQ reasoning by isolating and comparing a model’s high confidence candidates, rather than expanding the reasoning space through sampling or search.
2. We introduce a stability-oriented perspective on multiple-choice reasoning, showing that many errors arise from preference instability among plausible options rather than missing knowledge or insufficient reasoning steps.
3. We present a transition-based analysis of model predictions across stages, providing diagnostic insight into when and why correct intermediate preferences fail to translate into correct final answers.
4. Through extensive experiments across commonsense, educational, and mathematical benchmarks, we demonstrate that IoT consistently improves accuracy over strong baselines while incurring minimal computational overhead.

2 Inclusion of Thoughts

2.1 Methodology

Inclusion-of-Thoughts operationalizes structured self-filtering through a three-stage pipeline to enhance the performance of Multiple-Choice Question (MCQ) answering. In fact, IoT adopts a stability-oriented view of MCQ reasoning in which the goal is not to generate additional reasoning paths, but to enhance MCQ performance by testing whether a model’s highest-confidence preferences remain consistent under controlled option perturbations. IoT operates in three main stages:

Stage 1: Initial Preference Elicitation Given an MCQ with the original option set $O = \{o_1, \dots, o_n\}$, the model is first prompted using a standard CoT approach to generate its most probable answer, which we denote as the top-choice candidate $o_1^* \in O$. This initial output reflects the model’s raw preference over the full set.

Stage 2: Second Plausibility Assessment To probe beyond this surface preference and identify the next-best alternative, the option set is *perturbed*. We remove the initial (stage 1) selection o_1^* from O

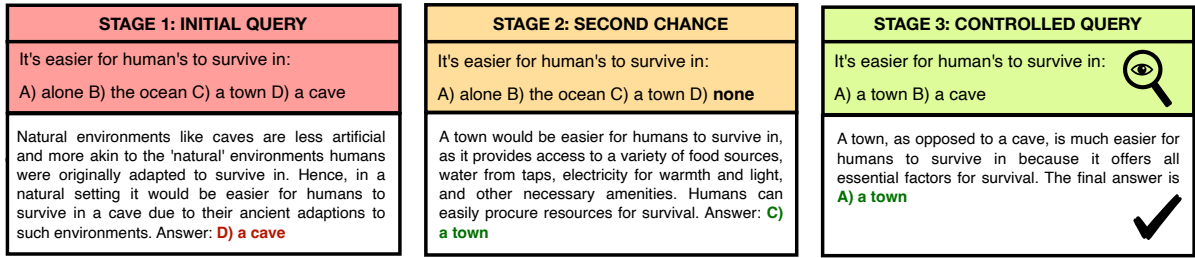


Figure 2: **Inclusion-of-Thoughts Framework.** The pipeline allows the model to choose up to two options. Then the model looks at its preferences and decides the final answer in isolation.

and replace it with a neutral placeholder, “*none of the options*”, yielding the modified option set

$$O' = (O \setminus \{o_1^*\}) \cup \{\text{none of the options}\}.$$

The model is then queried again on this modified MCQ, resulting in the second (stage 2) selection $o_2^* \in O'$. If o_1^* was a robustly correct answer, the model is expected to select its placeholder, “*none of the options*”. Otherwise, the model will select another plausible alternative. The resulting pair of answers, (o_1^*, o_2^*) , reveals a structured preference ordering.

Stage 3: Confined Final Inference The framework then constructs a reduced MCQ consisting solely of the two most plausible model-selected candidates. If $o_2^* = \text{“none of the options”}$, the reduced set is $\{o_1^*\}$ and causes early stopping (see Section 2.2); otherwise, the set is $\{o_1^*, o_2^*\}$. The final inference is performed over this minimally distracting set. This reframed question sharply reduces the model’s *cognitive load* (i.e. model’s preference instability in the presence of distractor options) and focuses the reasoning process on its own shortlisted alternatives. The final selection on this reduced set is taken as the IoT prediction, providing insight into the stability and robustness of the model’s internal reasoning.

2.2 Early Stopping Mechanism

Our framework incorporates a critical early stopping mechanism that terminates the process immediately if the most plausible option selected during **Stage 1** (o_1^*) is affirmed as the only meaningful choice in **Stage 2** (i.e., $o_2^* = \text{“none of the options”}$)—bypassing the Stage 3 prompting. When the model consistently signals that its initial choice is superior to all remaining options, the reasoning trajectory is considered converged, establishing a definitive final answer based on the confirmed

stability of its preference. An example of early stopping is presented in Figure 9.

3 Experiments

Our experimental evaluation focuses on three primary objectives: determining whether IoT delivers consistent accuracy improvements across diverse domains and model scales, comparing its computational efficiency and robustness against existing test-time methods, and diagnosing the specific types of reasoning failures it successfully mitigates versus those where it remains ineffective.

3.1 Setup

Tasks and Datasets To comprehensively evaluate the effectiveness of our proposed IoT framework, we conducted experiments on various MCQ benchmarks. These datasets span diverse domains and difficulty levels, enabling a robust validation of the model’s performance and generalization capabilities across a variety of challenging tasks. <https://aclanthology.org/2025.acl-long.1051.pdf>

- **Commonsense Reasoning** CommonsenseQA (CSQA, Talmor et al., 2019). A dataset emphasizing commonsense reasoning, with questions in a 5-option MCQs format. OpenbookQA (OBQA, Mihaylov et al., 2018). This dataset mimics open-book exams by pairing elementary science questions with a set of core facts. Answering requires multi-step reasoning that combines these provided facts with broad commonsense knowledge. SocialIQA (SIQA, Sap et al., 2019). This dataset is designed to test social commonsense reasoning about people’s actions, motivations, and emotional reactions in everyday scenarios.
- **Educational Benchmark** Massive Multi-Task Language Understanding (MMLU, Hendrycks et al., 2021), focuses on professional domains

such as law, medicine, engineering, and finance. AI2 Reasoning Challenge (ARC, Clark et al., 2018). Derived from U.S. elementary and middle school science exams, it is primarily used to assess the model’s performance in scientific knowledge and complex reasoning tasks.

- **Math Reasoning** Algebra Question Answering with Rationales (AQuA, Ling et al., 2017). This dataset contains algebra questions, each accompanied by detailed solution steps. It aims to evaluate the model’s problem-solving and logical reasoning abilities in mathematical tasks. GSM8K-MC. This dataset is an MCQ adaptation of the original GSM8K (Cobbe et al., 2021) dataset, containing elementary-level mathematics problems designed to assess the model’s mathematical reasoning capabilities.

Baselines We compared the performance gains of the IoT framework against the following baselines.

- **Chain-of-Thoughts** (CoT, Wei et al., 2023) is a technique where a language model is prompted to generate a series of short sentences that mimic the reasoning process a person might employ in solving a task.
- **Self-Consistency** (SC, Wang et al., 2023) replaces the naive greedy decoding used in chain-of-thought prompting. It first samples a diverse set of reasoning paths instead of only taking the greedy one, and then selects the most consistent answer by marginalizing out the sampled reasoning paths.
- **Exclusion of Thoughts** (EoT, Fu et al., 2025) uses a scoring approach to reduce cognitive load by eliminating the least important options step-by-step until the likelihood of the top option is higher than the other options with respect to a heuristic threshold.

3.2 Main Results

We comprehensively evaluated the proposed IoT method across various LLMs and compared its performance with mainstream baseline approaches. As demonstrated in Table 1, IoT achieves consistent accuracy improvements across most benchmarks, attaining the highest average accuracy in each category compared to existing baselines. For instance, when applied to the O1mo-2-7B model, IoT secures the highest overall average accuracy

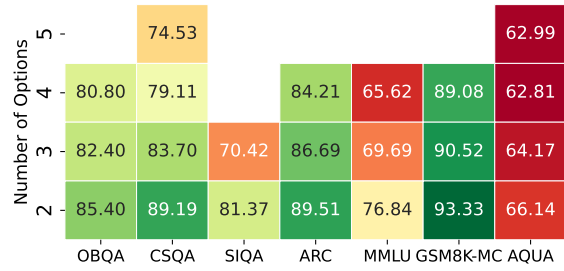


Figure 3: The figure illustrates the effect of removing options on the accuracy of the O1mo-2-7B on different datasets.

at 72.91%, surpassing its nearest competitor by a margin of approximately 1.64%.

The results also reveal specific domain nuances where IoT’s relative gains vary. IoT is observed to be less effective when applied to the SIQA dataset. This is attributed to SIQA’s 3-option format; because the framework filters down to the top-two plausible answers, the reduction in cognitive load is less significant than in benchmarks with larger option sets.

Furthermore, while IoT yields performance improvements over standard CoT on Math benchmarks, it is occasionally less effective than SC. This suggests that models often reason over mathematical questions independently of the provided options, as there is a direct logical pathway from the question to the solution. In such cases, SC benefits from allowing diverse reasoning pathways through temperature-based sampling, whereas IoT—along with other baselines—utilizes a greedy approach that prohibits such diversity. Despite these cases, IoT remains the most robust overall method, establishing the highest average performance across the O1mo-2-13B and Llama-3.3-8B models. Figure 4 illustrates the performance comparison of IoT with existing baselines across general domains.

Across individual tasks, IoT exhibits its strength, particularly in more complex reasoning domains. For instance, on the GSM8K-MC Math benchmark, IoT achieves the best result of 91.66 (O1mo-2-7B) and 93.10 (O1mo-2-13B), outperforming both the standard Chain-of-Thought (CoT) and Self-Consistency (SC) baselines. Similarly, in the Commonsense and Education domains, IoT consistently establishes the highest accuracy for the majority of datasets, including ARC, MMLU, and SIQA, confirming that its structured, self-filtering approach effectively enhances the model’s ability to select the most robust answer across diverse

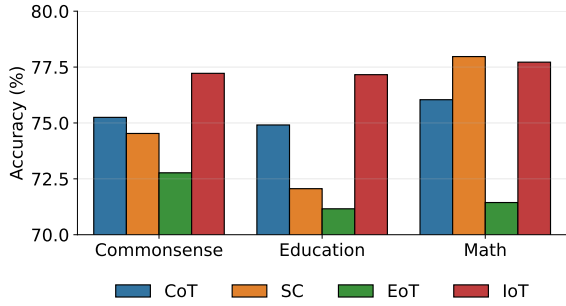


Figure 4: Results of 01mo-2-7B categorized by benchmarks type. While the margin in Commonsense or Education is more significant, IoT achieves performance comparable to self-consistency (SC) with much lower computational cost.

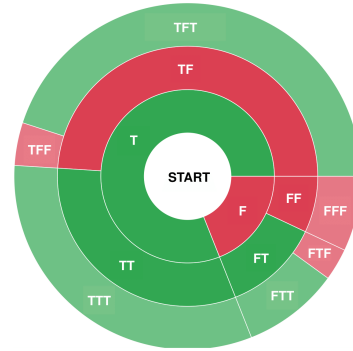


Figure 5: Answer transition process of 01mo-2-7B over OBQA dataset. The final improvement is the difference between FTT and TFF.

knowledge and reasoning tasks.

3.3 Transition Analysis

Beyond aggregate accuracy, IoT enables a fine-grained analysis of how model predictions evolve under option perturbation, revealing distinct transition patterns that characterize different reasoning failure modes. During each stage, the model output can be either True (*i.e.* it matches the groundtruth) or False (F). Therefore, from stage 1 to stage 3, the model answer can flip multiple times, and it can be categorized into the following groups:

- **TTT**: When the model detects and retains its correct answer during all stages, it indicates that the model has high confidence in the correct answer.
- **FFF**: In some cases, the model is not able to detect the correct answer in stages 1 and 2, and therefore stage 3. In these cases, the model has no knowledge about the answer.
- **TFT, FTT**: In these cases, the correct answer appears in either stage 1 or 2, and the model successfully chose the correct answer in stage 3.
- **TFF, FTF**: In these cases, the correct answers appear in either stage 1 or 2. However, during stage 3, the model chose the False option as the final answer.

We analyse this transition solely to justify the final answer. As shown in Figure 5, the performance improvement of IoT (outermost layer) w.r.t. CoT (innermost layer) is due to transforming some of the initially incorrect answers into correct answers (FTT) minus a small portion of correct answers turning into false (TFF).

The TFF and FTF cases are of critical importance for our analysis. These instances are characterized by the model identifying the correct answer within its intermediate steps but selecting an incorrect final prediction. Specifically, we hypothesize that when the model ranks the correct answer among its top two priorities, yet ultimately outputs the incorrect selection, this suggests one of two potential issues: (1) the presence of ambiguity indicating multiple plausible answers for the given input, or (2) an erroneous ground-truth label in the dataset. Below, we show an example of each case:

FTF - Multiple groundtruth

She was always helping at the senior center, it brought her what? (A) **happiness** (B) satisfaction

FTF - Wrong groundtruth

What could committing murder prevent someone from doing? (A) **find god** (B) go to jail

Ambiguous Cases To assess these ambiguous samples (TFF and FTF), we employed Gemini-3-pro and GPT-5 as external judges. Specifically, we use stage 3 questions with two high-priority options and ask the judges to select the correct answer. The rate at which the judges' selected answer aligns with the IoT final output is recorded as the agreement rate. As shown in Figure 6, the judges' evaluation suggests a flip in a portion of FTF and TFF cases, leading to transitions of FTF→TFT and TFF→FTT (in the case of multiple groundtruth: FTF/TFF→TTT). This suggests that IoT can serve as a diagnostic tool to systematically identify cases in which the benchmark's single-answer assumption may be insufficient to capture genuine ambiguity in the question.

Model	Method	Commonsense			Education		Math		Avg.
		OBQA	CSQA	SIQA	ARC	MMLU	GSM8K-MC	AQUA	
Olmoe-2-7B	CoT	<u>80.80</u>	74.53	70.42	<u>84.21</u>	<u>65.62</u>	89.08	62.99	<u>75.38</u>
	SC	75.80	<u>76.42</u>	71.36	79.22	64.91	<u>89.39</u>	66.54	74.81
	EoT	74.60	73.79	69.91	79.69	62.62	87.06	55.82	71.93
	IoT	84.20	76.58	<u>70.88</u>	87.54	66.78	91.66	<u>63.78</u>	77.35
Olmoe-2-13B	CoT	<u>85.40</u>	79.69	71.60	85.32	<u>70.59</u>	91.36	67.32	78.75
	SC	82.00	80.18	<u>72.93</u>	<u>86.01</u>	69.86	95.68	73.62	<u>80.04</u>
	EoT	83.40	73.38	72.57	85.58	69.62	90.14	58.23	76.13
	IoT	87.60	<u>80.01</u>	73.64	88.65	71.34	<u>93.10</u>	<u>70.47</u>	80.69
Llama-3.3-8B	CoT	74.60	72.89	69.75	81.06	65.56	87.41	54.72	72.28
	SC	<u>77.80</u>	<u>74.77</u>	71.14	<u>82.57</u>	69.00	91.89	<u>57.09</u>	<u>74.89</u>
	EoT	75.80	74.44	<u>70.57</u>	81.99	<u>66.56</u>	84.31	53.01	72.38
	IoT	78.40	75.84	69.70	83.79	66.31	<u>90.67</u>	61.81	75.22
GPT-4o mini	CoT	<u>88.80</u>	<u>83.21</u>	<u>79.52</u>	<u>94.54</u>	<u>81.31</u>	<u>94.74</u>	<u>74.37</u>	<u>85.23</u>
	IoT	93.40	84.54	80.04	94.62	82.06	95.00	79.13	86.21

Table 1: Accuracy of multiple LLMs across various benchmarks using (1) CoT, (2) Self-Consistency (SC), (3) stepwise elimination using EoT, (4) detecting plausible answers and reconstructing the question using IoT. The best performance for each task is shown in **bold**. The second best performance is underlined. **Note:** EoT cannot be applied directly to closed-source models (GPT), and GPT-4o SC is excluded due to computational costs.

388 Furthermore, this potential correction implies that
389 the actual performance of the IoT is better than the
390 measured performance, assuming that the identified
391 ambiguous or erroneous samples are corrected.

392 3.4 More Chances, Less Appreciation

393 To determine the optimal number of chances that
394 maximizes the performance, we extended our
395 framework to allow the model a potential oppor-
396 tunity at option selection. This modified process
397 filters the candidate options down to a maximum
398 of three distinct choices (or fewer, if the model re-
399 peats previous selections). The final question is
400 then constructed using these filtered options, and
401 the model is prompted for its definitive preference.

402 As demonstrated in Table 3, model performance
403 peaks when the selection process is limited to a sec-
404 ond chance, and degrades when a third chance is in-
405 troduced. This diminishing return can be attributed
406 to two main factors: (1) If the correct answer is
407 not successfully identified within the top two pre-
408 dictions, its inclusion in the third-ranked option
409 set suggests a significantly lower initial confidence.
410 The model’s ability to identify the correct answer
411 in this three-option context is thus considerably
412 impaired compared to its highly confident top two

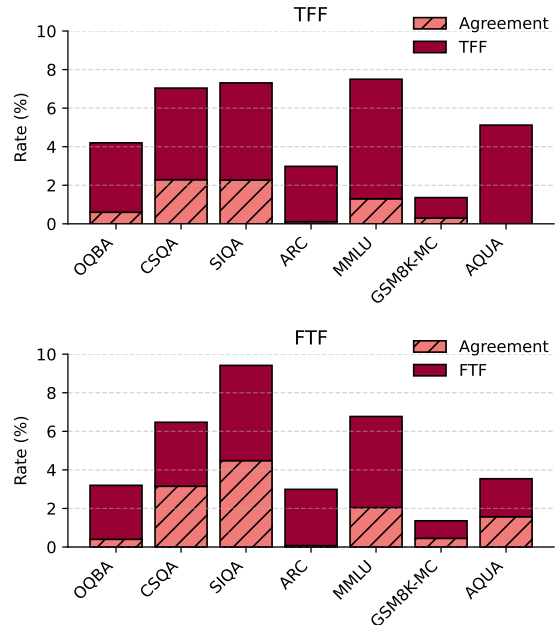


Figure 6: The portion of noisy samples agreed across different benchmarks in TFF and FTF agreed by external judges. The Agreement rate shown in the plot is the average rate over Gemini-3-pro, and GPT-5 agreement.

Dataset	CoT		IoT	
	Mean (\uparrow)	Std (\downarrow)	Mean (\uparrow)	Std (\downarrow)
OBQA	80.64	0.84	84.28	0.73
CSQA	72.64	1.13	76.89	0.52
SIQA	69.63	0.49	71.89	0.61
ARC	83.43	0.50	86.82	0.53
MMLU	64.58	0.64	66.84	0.29
GSM8K-MC	90.19	0.74	92.70	0.64
AQUA	58.11	3.53	63.93	2.02

Table 2: Robustness of IoT compared to CoT w.r.t. shuffling options. Best performance (highest mean, lowest std) is **bold**.

413 selections. (2) Increasing the number of selection
414 opportunities inherently increases the number of
415 options presented in the final, constructed question
416 (i.e., moving from two to three options). A greater
417 number of choices in the final reasoning stage in-
418 troduces greater complexity and a greater potential
419 for distraction, hindering the model’s capacity to
420 focus and select the correct answer effectively.

421 3.5 Robustness

422 [Pezeshkpour and Hruschka \(2023\)](#) reveal that
423 LLMs exhibit a “sensitivity gap,” where model pre-
424 dictions shift based solely on the positional order of
425 choices in multiple-choice benchmarks. This lack
426 of robust reasoning suggests that models often rely
427 on positional heuristics rather than a deep semantic
428 understanding of the options.

429 To ensure IoT provides a reliable measure of
430 task performance, we conduct a robustness analy-
431 sis by systematically shuffling the order of options
432 5 times and calculating the variance in performance
433 across permutations. We aim to quantify the extent
434 to which our framework mitigates these ordering
435 effects and maintains a consistent, label-invariant
436 representation of the underlying knowledge. Ta-
437 ble 2 results indicate that IoT not only improves
438 average accuracy but also reduces sensitivity to op-
439 tion ordering, suggesting a more stable underlying
440 preference structure.

441 3.6 Computational Overhead

442 Despite the iterative nature of the IoT, it introduces
443 only modest computational overhead in practice.
444 This efficiency is primarily achieved through a dy-
445 namic early-stopping mechanism (see Section 2.2).
446 If the most plausible option selected in Stage 1 (ini-
447 tial assessment) is identical to the option selected
448 in Stage 2 (initial filtering and reasoning), the rea-

Dataset	CoT	Inclusion of Thoughts	
		2nd Chance	3rd Chance
OBQA	77.70	81.30	80.70
CSQA	73.34	76.41	75.67
SIQA	70.09	70.29	N/A
ARC	82.64	85.67	84.81
MMLU	65.59	66.54	66.92
GSM8K-MC	88.25	91.17	89.50
AQUA	58.86	62.80	60.04

Table 3: Comparison of IoT performance given a sec-
ond and third chance. The results are average over
O1mo-2-7B and Llama-3. 3-8B performance.

Dataset	CoT	SC	EoT	IoT
OBQA	173	833	360	369
CSQA	159	712	326	427
SIQA	164	797	335	359
ARC	200	1,000	404	494
MMLU	261	1,336	529	636
GSM8K-MC	162	895	340	258
AQUA	279	1,468	604	430
Average	218	1,005	449	488

Table 4: Average tokens generated per sample. SC
generates roughly $4.6\times$ the tokens of CoT, and EoT
generates $2.1\times$, while IoT is approximately $2.2\times$
the cost of CoT.

449 soning process is deemed to have converged. In
450 this case, the Stage 3 prompting step, which re-
451 quires further token generation, is automatically
452 bypassed. This immediate termination conserves
453 computational resources by preventing redundant
454 reasoning steps when the model has already estab-
455 lished a high-confidence, consistent answer. To em-
456 pirically validate this, we evaluated the framework
457 on O1mo-2-7B and measured the total number of
458 tokens generated by IoT across various tasks, com-
459 pared with baseline methods. Table 4 indicates that
460 the extra computational cost introduced by IoT is
461 limited, providing a favorable trade-off given the
462 substantial performance gains it yields in complex
463 reasoning tasks.

464 4 Related Works

465 **Reasoning with LLMs** The ability of large lan-
466 guage models (LLMs) to generate intermediate
467 steps, often termed a Chain-of-Thought (CoT) ([Wei
468 et al., 2023](#)), has become a central focus in rea-
469 soning research. Several works have explored the
470 internal mechanics of CoT, probing its faithfulness
471 ([Lyu et al., 2023](#); [Lanham et al., 2023](#)) and its util-

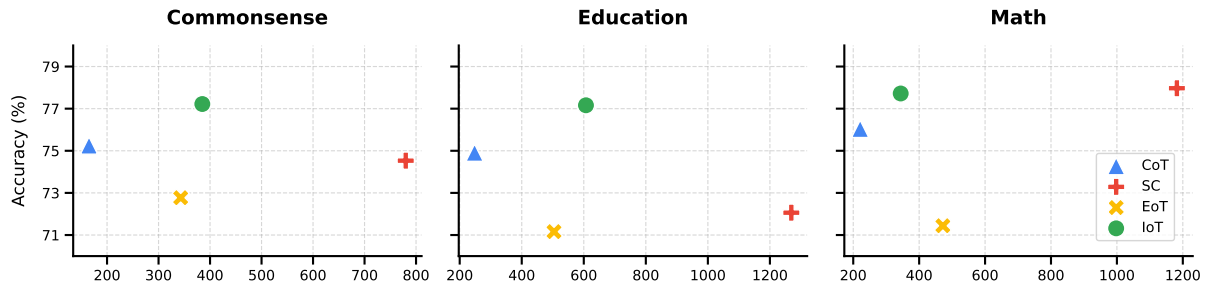


Figure 7: Cost-Benefit. Vertical axes illustrate the accuracy of each method, while the horizontal axes show the average number of tokens per sample that each method costs.

ity in complex planning (Saparov and He, 2022). Further prompt engineering techniques have been developed to enhance this process by encouraging the model to consider multiple viewpoints or generate supporting evidence for competing options, such as Debate Prompting (Michael et al., 2023) and Maieutic Prompting (Jung et al., 2022). Prior to the rise of general CoT methods, research largely focused on specialized approaches for tasks like mathematical reasoning (Andor et al., 2019; Ran et al., 2019; Geva et al., 2020; Piękos et al., 2021). In contrast, the introduction of Self-Consistency (SC) by Wang et al. (2023) provided a more generalized framework, applicable to a wide range of reasoning tasks without requiring additional supervision or fine-tuning, thereby substantially improving the effectiveness of standard CoT.

Test-time Scaling A large body of research has shown that LLM reasoning can be dramatically improved via advanced test-time approaches that combine prompting and aggregation. Representative methods include the aforementioned CoT and SC, as well as Tree-of-Thought prompting (Yao et al., 2023b), and Self-Reflection mechanisms (Shinn et al., 2023b; Madaan et al., 2023). Furthermore, techniques that leverage backward reasoning have been proposed to verify intermediate steps and improve precision, particularly in mathematical contexts (Weng et al., 2023; Xue et al., 2023; Jiang et al., 2024). Orthogonal to search- or verification-heavy test-time scaling, recent analyses have shown that LLM predictions can be highly sensitive to superficial perturbations such as option ordering or minor context changes, indicating unstable preference structures rather than reasoning deficiencies (Pezeshkpour and Hruschka, 2024). These findings suggest that improving inference-time robustness may require mechanisms that explicitly probe and stabilize model preferences under con-

trolled perturbations. While these advanced test-time methods are effective, particularly compared to baseline CoT, they often yield only moderate additional improvement over simpler techniques such as SC alone (Wang et al., 2023). Crucially, many of these highly complex approaches, such as backward verification and specific multi-agent setups, have been primarily developed and validated for mathematical tasks, which inherently limits their generalizability to the broader linguistic and commonsense challenges targeted by our framework. Our proposed IoT framework distinguishes itself by focusing on improving the selection mechanism when multiple plausible outputs are already generated, thereby maximizing the utility of the model’s intermediate high-priority outputs without introducing complex, task-specific reasoning steps. Unlike prior methods that expand or verify reasoning trajectories, IoT treats preference instability itself as the primary failure mode and addresses it through structured option-level perturbation, enabling targeted improvement without increasing reasoning depth or search complexity.

5 Conclusion

We introduced the IoT framework, a novel prompting strategy that systematically eliminates incorrect options to enhance LLM performance on MCQ tasks. Inspired by human reasoning strategies, IoT reduces cognitive load by redirecting the model’s attention to critical options, allowing it to focus more effectively on analyzing the relevant choices. Our extensive experiments across a variety of MCQ datasets demonstrate that IoT improves reasoning accuracy, particularly for challenging tasks. We believe that viewing MC reasoning through the lens of preference stability opens new directions for understanding and improving LLM reasoning behavior under controlled perturbations.

549 Limitations

550 IoT is designed specifically for MC question an-
551 swering and may not directly generalize to open-
552 ended generation tasks. Its effectiveness dimin-
553 ishes when the option set is very small, as prefer-
554 ence instability is less pronounced in such settings.
555 Finally, while IoT improves stability at inference
556 time, it does not address errors caused by missing
557 knowledge or fundamentally incorrect reasoning.

558 Ethics Statement

559 This research presents no ethical concerns. All
560 experiments were conducted using publicly avail-
561 able datasets exclusively for research purposes. We
562 carefully reviewed the selected datasets to ensure
563 they do not contain unethical content, private infor-
564 mation, or sensitive topics. The foundation models
565 used in this study are openly accessible and have
566 been employed in accordance with their research-
567 oriented licenses. Moreover, AI assistants were
568 utilized in the writing and refinement of this paper.

569 References

570 AI@Meta. 2024. [Llama 3 model card](#).

571 Daniel Andor, Luheng He, Kenton Lee, and Emily
572 Pitler. 2019. [Giving bert a calculator: Finding oper-
573 ations and arguments with reading comprehension](#).
574 *Preprint*, arXiv:1909.00109.

575 Nishant Balepur, Shramay Palta, and Rachel Rudinger.
576 2024. [It’s not easy being wrong: Large language
577 models struggle with process of elimination reason-
578 ing](#). In *Findings of the Association for Computa-
579 tional Linguistics: ACL 2024*, pages 10143–10166.

580 BIG bench authors. 2023. [Beyond the imitation game:
581 Quantifying and extrapolating the capabilities of lan-
582 guage models](#). *Transactions on Machine Learning
583 Research*.

584 Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot,
585 Ashish Sabharwal, Carissa Schoenick, and Oyvind
586 Tafjord. 2018. [Think you have solved question
587 answering? try arc, the ai2 reasoning challenge](#).
588 *Preprint*, arXiv:1803.05457.

589 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian,
590 Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias
591 Plappert, Jerry Tworek, Jacob Hilton, Reiichiro
592 Nakano, Christopher Hesse, and John Schulman.
593 2021. [Training verifiers to solve math word prob-
594 lems](#). *Preprint*, arXiv:2110.14168.

595 Qihang Fu, Yongbin Qin, Ruizhang Huang, Yanping
596 Chen, Yulin Zhou, and Lintao Long. 2025. [Exclu-
597 sion of thought: Mitigating cognitive load in large](#)

[language models for enhanced reasoning in multiple-
choice tasks](#). In *Proceedings of the 63rd Annual
Meeting of the Association for Computational Lin-
guistics (Volume 1: Long Papers)*, pages 21673–
21686, Vienna, Austria. Association for Computa-
tional Linguistics.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. [Injecting numerical reasoning skills into language models](#). *Preprint*, arXiv:2004.04487.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. [Reasoning with language model is planning with world model](#). *Preprint*, arXiv:2305.14992.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. [Measuring massive multitask language understanding](#). *Preprint*, arXiv:2009.03300.

Weisen Jiang, Han Shi, Longhui Yu, Zhengying Liu, Yu Zhang, Zhenguo Li, and James Kwok. 2024. [Forward-backward reasoning in large language models for mathematical verification](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6647–6661.

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. [Maieutic prompting: Logically consistent reasoning with recursive explanations](#). In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 1266–1279.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, and 1 others. 2023. [Measuring faithfulness in chain-of-thought reasoning](#). *arXiv preprint arXiv:2307.13702*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. [Program induction by rationale generation : Learning to solve and explain algebraic word problems](#). *Preprint*, arXiv:1705.04146.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. [Faithful chain-of-thought reasoning](#). In *The 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AAACL 2023)*.

Chenkai Ma and Xinya Du. 2023. [Poe: Process of elimination for multiple choice reasoning](#). *Preprint*, arXiv:2310.15575.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder,

598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652

653	Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback . <i>Preprint</i> , arXiv:2303.17651.	Abulhair Saparov and He He. 2022. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. <i>arXiv preprint arXiv:2210.01240</i> .	709
654			710
655			711
656			
657	Julian Michael, Salsabila Mahdi, David Rein, Jackson Petty, Julien Dirani, Vishakh Padmakumar, and Samuel R Bowman. 2023. Debate helps supervise unreliable experts. <i>arXiv preprint arXiv:2311.08702</i> .	Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023a. Reflexion: Language agents with verbal reinforcement learning . <i>Preprint</i> , arXiv:2303.11366.	712
658			713
659			714
660			715
661	Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering . <i>Preprint</i> , arXiv:1809.02789.	Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023b. Reflexion: Language agents with verbal reinforcement learning . <i>Advances in Neural Information Processing Systems</i> , 36:8634–8652.	716
662			717
663			718
664			719
665	Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2024. olmo 2 furious .	Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge . <i>Preprint</i> , arXiv:1811.00937.	721
666			722
667			723
668			724
669			
670			725
671			726
672	OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card . <i>Preprint</i> , arXiv:2410.21276.	Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models . <i>Preprint</i> , arXiv:2203.11171.	727
673			728
674			729
675			
676			730
677			731
678			732
679	Pouya Pezeshkpour and Estevam Hruschka. 2023. Large language models sensitivity to the order of options in multiple-choice questions . <i>Preprint</i> , arXiv:2308.11483.	Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models . <i>Preprint</i> , arXiv:2201.11903.	733
680			734
681			735
682			736
683	Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In <i>Findings of the Association for Computational Linguistics: NAACL 2024</i> , pages 2006–2017.	Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Iliia Kulikov, and Zaid Harchaoui. 2024. From decoding to meta-generation: Inference-time algorithms for large language models . <i>Preprint</i> , arXiv:2406.16838.	737
684			738
685			739
686			
687			740
688	Piotr Piękos, Henryk Michalewski, and Mateusz Malinowski. 2021. Measuring and improving bert’s mathematical abilities by predicting the order of reasoning . <i>Preprint</i> , arXiv:2106.03921.	Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification. In <i>Findings of the Association for Computational Linguistics: EMNLP 2023</i> , pages 2550–2575.	741
689			742
690			743
691			744
692	Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, and 61 others. 2022. Scaling language models: Methods, analysis insights from training gopher . <i>Preprint</i> , arXiv:2112.11446.	Tianci Xue, Ziqi Wang, Zhenhailong Wang, Chi Han, Pengfei Yu, and Heng Ji. 2023. Rcot: Detecting and rectifying factual inconsistency in reasoning by reversing chain-of-thought . <i>arXiv preprint arXiv:2305.11499</i> .	745
693			746
694			747
695			748
696			749
697			750
698			
699			751
700			752
701	Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. 2019. Numnet: Machine reading comprehension with numerical reasoning . <i>Preprint</i> , arXiv:1910.06701.	Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models . <i>Preprint</i> , arXiv:2305.10601.	753
702			754
703			755
704			
705	Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialliqa: Commonsense reasoning about social interactions . <i>Preprint</i> , arXiv:1904.09728.	Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023b. Tree of thoughts: Deliberate problem solving with large language models . <i>Advances in neural information processing systems</i> , 36:11809–11822.	756
706			757
707			758
708			759
		Eric Zhao, Pranjal Awasthi, and Sreenivas Gollapudi. 2025. Sample, scrutinize and scale: Effective inference-time search by scaling verification . <i>Preprint</i> , arXiv:2502.01839.	760
			761
			762
			763
			764

A Evaluation Prompt

Below, we provide the prompt used for evaluating LLMs.

Evaluation Prompt

You are given a multiple-choice question. You should reason in a step-by-step manner to get the right answer.

Question: {question} {options}

Judge Prompt

You are given a multiple-choice question and a reasoning text. Your task is to extract the answer choice according to the reasoning text. Output only the option letter.

Question: {question} {options}
Reasoning text: {reasoning}
Answer:

case highlights IoT’s core mechanism: isolating the model’s top plausible candidates and stabilizing the final decision by reducing distractor-induced ambiguity.

Figure 9 on page 13 shows an example where IoT terminates early after Stage 2, as the model confirms its initial preference by selecting the placeholder option (“none of the options”). This behavior corresponds to the TTT transition pattern described in Section 3.3. It demonstrates how IoT avoids unnecessary prompting when preference stability is already established, thereby reducing computational overhead without sacrificing accuracy.

B Datasets Statistics

Table 5 summarizes the data distribution for the benchmarks used in our experiments, illustrating the number of samples across the training, development, and testing splits.

Dataset	Train	Dev	Test
OBQA	7.69K	500	500
CSQA	9.74K	1.22K	1.14K
SIQA	33410	1954	-
ARC	1.12K	299	1.17K
MMLU	-	-	14K
GSM8K-MC	7.47K	-	1.32K
AQUA	5K	-	254

Table 5: Number of samples each dataset contains for train, dev, and test splits.

C Examples of Inclusion-of-Thoughts

To improve interpretability and transparency, we present representative step-by-step examples of the Inclusion-of-Thoughts (IoT) framework, illustrating both standard three-stage reasoning and early stopping behavior observed in our experiments.

Figure 8 on page 12 provides a step-by-step qualitative example of the IoT framework applied to an OBQA sample. The example illustrates how an initially correct but unstable preference in Stage 1 is challenged in Stage 2, and subsequently resolved through a confined comparison in Stage 3. This

STAGE 1

A cactus stem is used to store (A) fruit (B) liquid (C) food (D) spines.

Step 1: Identify the properties of a cactus stem. - A cactus stem is a thick, fleshy structure that can store water. Step 2: Analyze the options given in the question. - (A) Fruit: Fruit is typically associated with flowering plants and does not store water in the same way a cactus stem does. - (B) Liquid: A cactus stem can store liquid, specifically water. - (C) Food: Food is typically the product of photosynthesis or consumption by organisms and is not stored in the same way as water in a cactus stem. - (D) Spines: Spines are sharp, rigid structures used for protection and do not store water. Step 3: Determine which option matches the function of a cactus stem. - The correct answer is (B) liquid, as a cactus stem is designed to store water. Step 4: Select the final answer. - **The final answer is (B).**

STAGE 2

A cactus stem is used to store (A) fruit (B) none of the options (C) food (D) spines.

Step 1: Identify the function of a cactus stem. - A cactus stem is used to store water. Step 2: Analyze the given options based on the function. - (A) Fruit: Storing fruit is not a function of a cactus stem. - (B) None of the options: This is a catch-all that doesn't provide a specific answer based on the stem's function. - (C) Food: Food is something that is consumed, not stored in a cactus stem. - (D) Spines: Spines are protective structures on a cactus, not something stored in the stem. Step 3: Determine the correct answer based on the function of the cactus stem. - The correct answer is (D) spines, as spines are stored on the cactus stem for protection. Therefore, **the final answer is (D).**

STAGE 3

A cactus stem is used to store (A) liquid (B) spines.

Step 1: Identify the function of a cactus stem. - A cactus stem is used to store water. Step 2: Compare the options with the identified function. - Option A: "liquid" is a form of water, which is the substance stored in a cactus stem. - Option B: "spines" are not a substance that can be stored in a cactus stem; they are protective structures. Step 3: Determine the correct answer. - **The correct answer is (A) liquid.**

Figure 8: Example of IoT (OBQA, Olmo-2-7b).

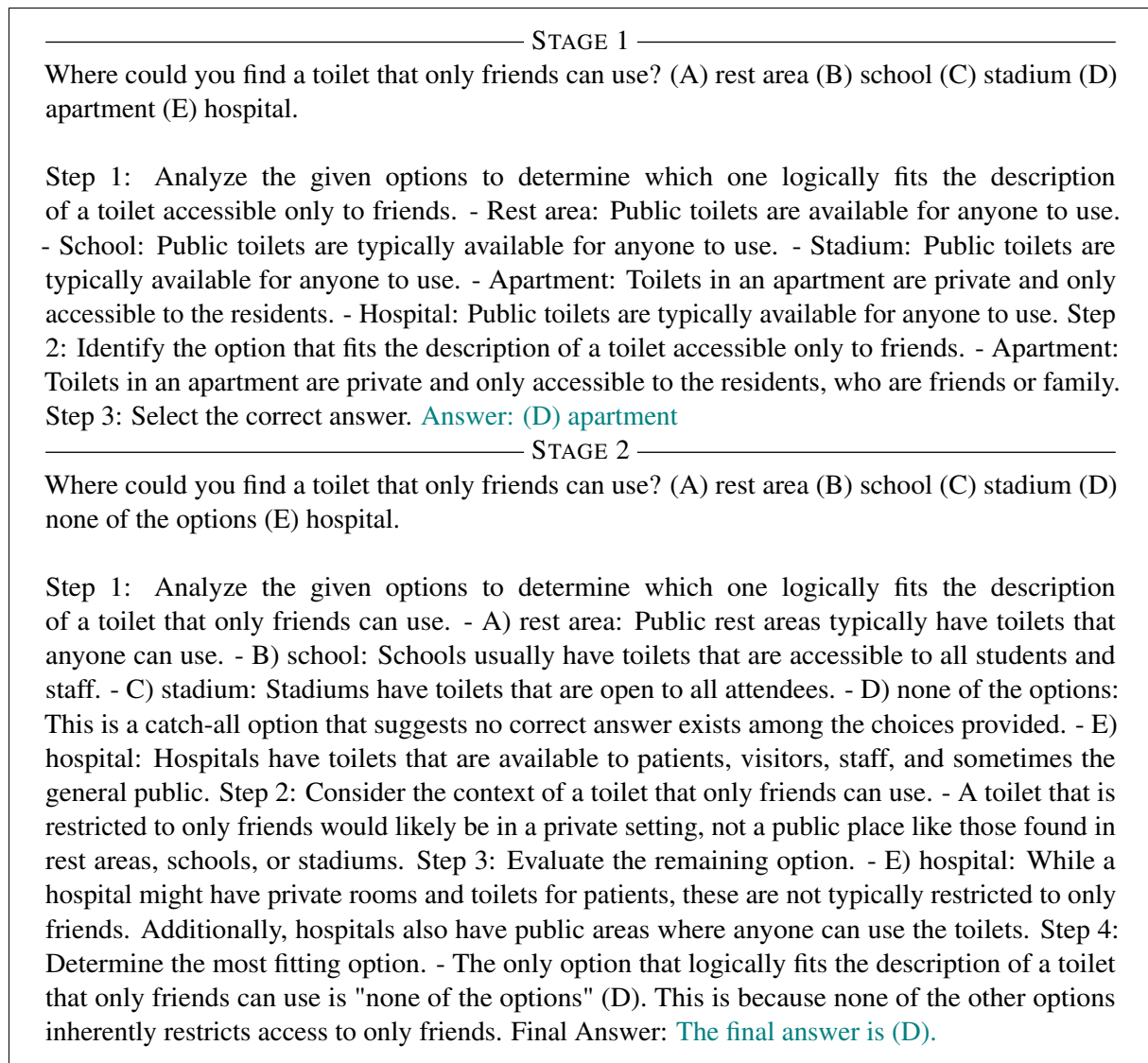


Figure 9: Example of IoT (CSQA, Olmo-2-7b). Here, the IoT stops the process and skips stage 3 since the model confirms its answer in stage 2 (TTT case).