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. They often divert 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. 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]

1 Introduction

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

Wei et al. (2023) propose chain-of-thought (CoT) prompting, where a large language model (LLM) is prompted to generate a series of short sentences that mimic the reasoning process a person might use in solving a task. While effective, a number of studies found that when CoT prompting fails, they are sometimes attributed to LLM's instability in decision making when exposed to multiple plausible distractors (Ma and Du, 2023; Fu et al., 2025; Balepur et al., 2024). That is, even though the correct answer has a high probability under the LLM, when presented with more options, the correct answer sees a consistent decrease of probability may result in the model changing its selection and therefore, reduction in the final performance (Figure 3) — we call this phenomenon *preference instability*.

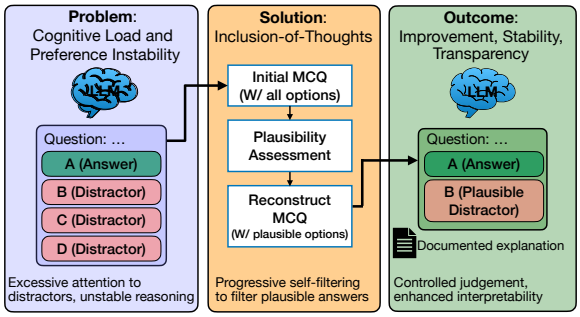

Figure 1: **Inclusion-of-Thoughts Framework** is a self-filtering strategy that reconstructs multiple-choice questions using only plausible options to mitigate model instability and enhance performance.

Guided by this insight, we introduce **Inclusion-of-Thoughts (IoT)**, a novel test-time approach designed to augment the standard CoT process by focusing on LLM's highest-ranked predictions. As illustrated in Figure 1, IoT achieves this by strategically *perturbing the option* to elicit and then isolate the model's top two preferences, followed by a final, two-choice 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 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 performance; it can also serve as a probing test to understand LLM behaviours. Another key advantage of IoT is its zero-shot, unsupervised implementation. It works with closed- or open-source LLMs, requiring no additional human annotation, auxiliary models, extra training, or fine-tuning.

We evaluate IoT on a wide range of arithmetic, commonsense reasoning, and ed-

[1] Code will be released once the paper is accepted.

ucational tasks over 4 LLMs of varying sizes: `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 LLMs, IoT improves over CoT prompting by a substantial margin across most tasks. In particular, `Olmo-2-7b` with IoT achieves surpass CoT across commonsense reasoning tasks (OBQA (Mihaylov et al., 2018) (+3.40%), CSQA (Talmor et al., 2019) (+2.05%), GSM8K-MC (Cobbe et al., 2021) (+3.95%) and educational benchmarks tasks (ARC (Clark et al., 2018) (+3.33%) and MMLU (Hendrycks et al., 2021) (+1.16%)). To summarise, our contributions are:

1. We propose IoT, a lightweight, zero-shot test-time framework that improves MCQ reasoning by isolating an LLM's high probability candidates rather than expanding the reasoning space through sampling or search.

2. We reveal that errors in MCQ reasoning often arise from preference instability among plausible options rather than missing knowledge or insufficient reasoning steps.

3. We present a transition-based analysis of LLM 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 Related Works

Reasoning with LLMs The ability of large language models (LLMs) to generate intermediate steps, a.k.a. Chain-of-Thought (CoT) (Wei et al., 2023), has become a central focus in reasoning research. Several works have explored the internal mechanics of CoT, probing its faithfulness (Lyu et al., 2023; Lanham et al., 2023) and its utility 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., 2023), and Self-Reflection mechanisms (Shinn et al., 2023; 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 controlled 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. 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 is unique in that it focuses on improving the selection mechanism when there are multiple plausible 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 tar-

geted improvement without increasing reasoning depth or search complexity.

3 Inclusion-of-Thoughts

3.1 Methodology

IoT operates in three stages:

Stage 1: Initial Preference Elicitation Given an MCQ with the original option set $O = \{o_1, \ldots, o_n\}$, the LM 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 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 simplified MCQ consisting only the two most plausible model-selected candidates. The final inference is performed over this minimal set, reducing the model's preference instability.

3.2 Early Stopping Mechanism

Our framework incorporates an early stopping mechanism that terminates the process immediately if the most plausible option selected during **Stage 1** (o_1^*) is confirmed in **Stage 2** (i.e., $o_2^* =$ *"none of the options"*). An example of early stopping is presented in Appendix F.

4 Experiments

Our 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 can (or cannot) mitigate.

4.1 Setup

Tasks and Datasets We conducted experiments on various MCQ benchmarks. These datasets span diverse domains and difficulty levels, allowing us to validate IoT's generalization across a variety of tasks. The datasets also have different numbers of options, and we can see how performance gain varies on this dimension.

- **Commonsense Reasoning** CommonsenseQA (CSQA, Talmor et al., 2019). A dataset on 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.

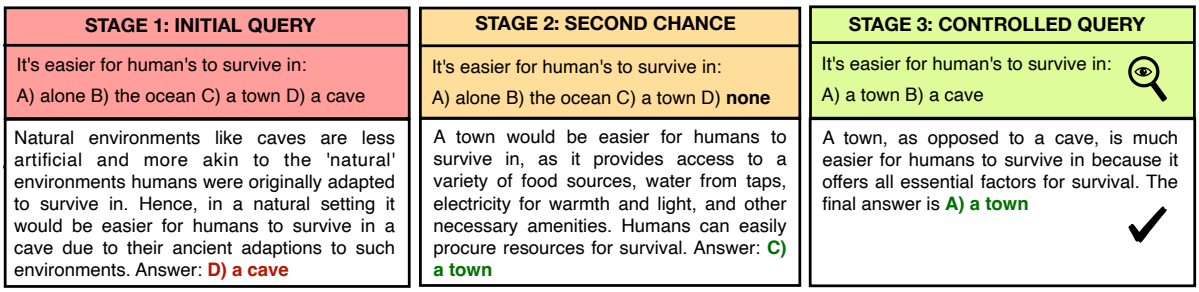

Figure 2: **Inclusion-of-Thoughts Framework.** The pipeline allows the model to choose up to two options. Then the model looks at its preferences and decides the final answer in isolation.

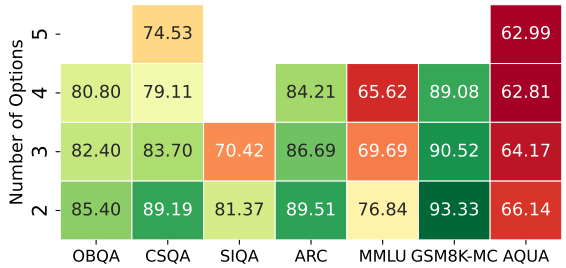

Figure 3: The figure illustrates the effect of options on the accuracy of `Olmo-2-7B` across different datasets. The y-axis indicates the number of options passed with the question.

Section 4.4 presents additional results with other baselines with comparable computational cost to IoT.

- **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 eliminate the least important options one at a time until the likelihood of the top option is higher than the other options with respect to a heuristic threshold.

While IoT shares a superficial similarity with elimination-based techniques, *e.g.* EoT, it is fundamentally distinguished by several key conceptual and procedural differences:

Focus on Preference Stability vs. Information Reduction Methods like EoT typically operate on the premise of reducing cognitive load by iteratively removing the least likely distractors based on heuristic scoring. In contrast, IoT probes the model's internal ranking through the counterfactual removal of its top-rated hypothesis. By isolating the top two candidates between which a model is most likely to oscillate, IoT treats "preference instability" as the primary failure mode rather than a lack of raw reasoning steps.

Computational Efficiency IoT is designed to be more lightweight than iterative elimination or complex search-based strategies. This efficiency is driven by a dynamic early-stopping mechanism. If the model selects the "none of the options" placeholder in Stage 2, it confirms its initial preference is robustly superior to all alternatives, allowing the process to terminate and bypass Stage 3. Our analysis shows IoT generates approximately 2.2x the tokens of standard CoT, whereas SC requires roughly 4.6x.

Diagnostic Utility Unlike prior methods, the three-stage structure of IoT enables a granular transition-based analysis. By categorizing model behavior into patterns such as TTT, FTT, or TFF, we can identify whether errors arise from model failure or external factors like genuine question ambiguity and "noisy" benchmark labels. This allows IoT to function as a diagnostic tool for assessing dataset quality in addition to improving performance.

4.2 Main Results

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

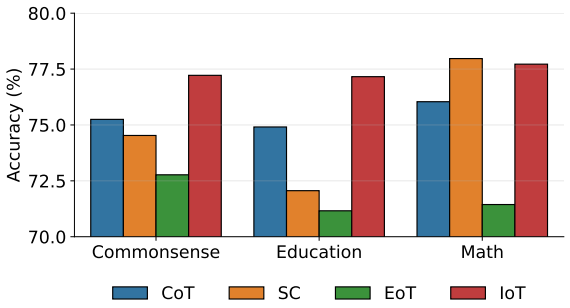

Figure 4: Results of `Olmo-2-7B` categorized by benchmarks type. While the margin in Commonsense or Education is more significant, IoT achieves performance comparable to self-consistency (SC) with much lower computational cost.

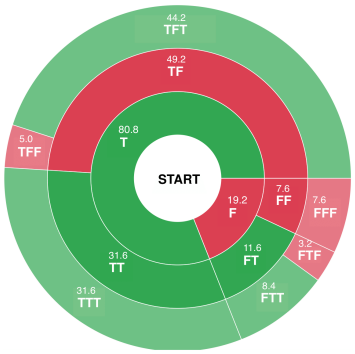

Figure 5: Answer transition process of `Olmo-2-7B` over OBQA dataset. The final improvement is the difference between FTT and TFF.

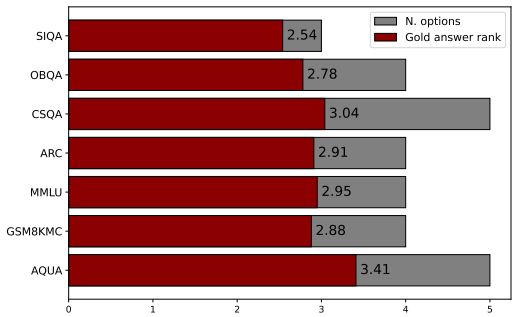

Figure 6: Gold answer average ranking for FFF cases.

instance, when applied to the `Olmo-2-7B` model, IoT secures the highest overall average accuracy 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 appears to be less effective when applied to the `SIQA` dataset. This is due to `SIQA`'s 3-option format. When the option set is smaller and therefore, distractor competition is low, models encounter less fluctuation, and improvements naturally shrink, as fewer competing hypotheses reduce preference instability.

While IoT yields performance improvements over standard CoT on Math benchmarks, it is less effective than SC. This suggests that for mathematical problems models maybe reasoning without considering the provided options. In such cases, SC facilitates diverse reasoning pathways through temperature-based sampling, whereas IoT (and other baselines) utilizes a greedy approach that doesn't promote such diversity. All that said, IoT remains the most robust overall method, establishing the highest average performance across all four models. Figure 4 illustrates the performance comparison of IoT with existing baselines across aggregated domain performance.

4.3 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 ground truth) 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.. Figure 6 shows that in FFF cases the gold answer does not appear within the model's top prediction, meaning that it has weak knowledge about the gold 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, IoT fails to recover the correct answer in stage 3.

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).

Model	Method	Commonsense			Education		Math		Avg.
		SIQA (3)	OBQA (4)	CSQA (5)	ARC (4)	MMLU (4)	GSM8K-MC (4)	AQUA (5)	
Olmo-2-7B	CoT	70.42	80.80	74.53	84.21	65.62	89.08	62.99	75.38
	SC	**71.36**	75.80	76.42	79.22	64.91	89.39	**66.54**	74.81
	EoT	69.91	74.60	73.79	79.69	62.62	87.06	55.82	71.93
	IoT	70.88	**84.20**	**76.58**	**87.54**	**66.78**	**91.66**	63.78	**77.35**
Olmo-2-13B	CoT	71.60	85.40	79.69	85.32	70.59	91.36	67.32	78.75
	SC	72.93	82.00	80.18	86.01	69.86	**95.68**	**73.62**	80.04
	EoT	72.57	83.40	73.38	85.58	69.62	90.14	58.23	76.13
	IoT	**73.64**	**87.60**	80.01	**88.65**	**71.34**	93.10	70.47	**80.69**
Llama-3.3-8B	CoT	69.75	74.60	72.89	81.06	65.56	87.41	54.72	72.28
	SC	**71.14**	77.80	74.77	82.57	**69.00**	**91.89**	57.09	74.89
	EoT	70.57	75.80	74.44	81.99	66.56	84.31	53.01	72.38
	IoT	69.70	**78.40**	**75.84**	**83.79**	66.31	90.67	**61.81**	**75.22**
GPT-4o mini	CoT	79.52	88.80	83.21	94.54	81.31	94.74	74.37	85.23
	IoT	**80.04**	**93.40**	**84.54**	**94.62**	**82.06**	**95.00**	**79.13**	**86.21**

Table 1: Accuracy of multiple LLMs across various benchmarks using CoT, Self-Consistency (SC), EoT, detecting plausible answers, and IoT. 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.

We next look into the TFF and FTF cases. These instances are interesting because the model identifies the correct answer at one stage but selects the incorrect answer in the end. We suspect this may be due to noisy ground truth: perhaps they are both plausible answers or the ground truth answer is incorrect. 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

To assess these cases (TFF and FTF) quantitatively, we use `Gemini-3-pro` and `GPT-5` as LLM judges. Specifically, we use stage 3 questions with two model-selected 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 7, the judges' evaluation suggests that perhaps a substantial portion of these "error cases" are perhaps not error after all (notwithstanding the variance across datasets) and that the performance we saw in Table 1 is a lower bound of its true performance.

4.4 Matched-budget Baselines

The simplicity of IoT is a major strength. To better contextualise this property, we compare IoT with two simple alternatives that have comparable computational cost to IoT: (1) top-1 + one random selection, which retains the highest-scoring option and one randomly selected alternative, thereby eliminating the computational cost of Stage 2; and (2) logit-based filtering, a simple heuristic that selects the two options with the highest likelihood scores.

Since Stage 3 is the same across all baselines, we only compare the top-2 accuracy. As shown in Table 9, IoT significantly improves top-2 accuracy compared to random second-option selection. Importantly, this improvement translates into higher final-stage accuracy, indicating that the gains are not merely due to additional opportunities but arise from structured counterfactual perturbation.

We also further evaluated SC with 3 samples (closer to IoT in token cost) for `Olmo-2-7b`. As can be seen in Table 3, IoT remains competitive or superior on most non-math tasks at similar budgets.

4.5 More Chances, Less Performance

To determine the optimal number of chances that maximizes the performance, we extend our framework to allow the model a potential opportunity at option selection. This modified process filters the candidate options down to a maximum of *three* distinct choices (or fewer, if the model repeats pre-

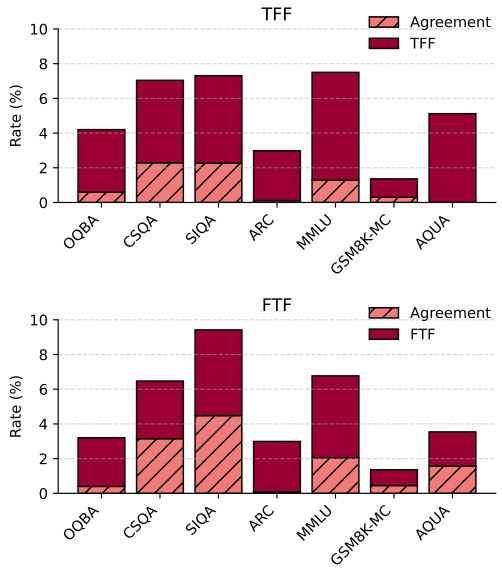

Figure 7: The portion of noisy samples agreed across different benchmarks in TFF and FTF agreed by external judges. The Agreement rate shown in the plot is the average rate over `Gemini-3-pro`, and `GPT-5` agreement.

Dataset	top-1+random	logit-based	IoT
SIQA	75.79	**90.78**	88.02
OBQA	83.00	86.80	**92.40**
CSQA	79.28	87.14	**89.19**
ARC	86.94	89.85	**93.86**
GSM8K-MC	91.89	58.68	**94.54**
AQUA	**74.02**	42.23	73.23

Table 2: Top-2 accuracy of match-budget baselines compared to IoT.

vious selections). The final question is then constructed using these filtered options.

As demonstrated in Table 4, model performance peaks when the selection process is limited to a second chance, and degrades when a third chance is introduced. This diminishing return can be attributed to two main factors: (1) If the correct answer is in the top two predictions, then it means it has a low model probability. And so the likelihood of the model recovering its performance in the final stage (when given 3 options) is low. (2) Increasing the number of option inherently increases the number of distractors in the final stage, and this can negatively impact model decision making (in other words, there is an opportunity cost for adding every additional candidate).

4.6 Robustness

Pezeshkpour and Hruschka (2023) reveal that

Dataset	CoT	SC (3 samples)	SC (5 samples)	IoT
SIQA	70.42	68.73	**71.36**	70.88
OBQA	80.80	80.80	75.80	**84.20**
CSQA	74.53	74.12	76.42	**76.58**
ARC	84.21	84.38	79.22	**87.54**
MMLU	65.62	N/A	64.91	**66.78**
GSM8K-MC	89.08	**93.32**	89.39	91.66
AQUA	62.99	65.35	**66.54**	63.78

Table 3: Comparison of self-consistency with different numbers of samples to IoT.

Dataset	CoT	Inclusion-of-Thoughts	
		w/ 2 Chances	w/ 3 Chances
SIQA	70.09	**70.29**	N/A
OBQA	77.70	**81.30**	80.70
CSQA	73.34	**76.41**	75.67
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 4: Comparison of IoT performance given a second and third chance. The results are average over `Olmo-2-7B` and `Llama-3.3-8B` performance.

LLMs exhibit a "sensitivity gap", where model predictions shift based on the positional order of choices in multiple-choice benchmarks. To ensure IoT provides a reliable measure of task performance, we conduct a robustness analysis by systematically shuffling the order of options 5 times and calculating the variance in performance across permutations. We also compute the selection entropy, where we measure how consistently the model stays locked onto the correct semantic choice versus getting pulled away by positional bias. We aim to quantify the extent to which our framework mitigates these ordering effects and maintains a consistent, label-invariant representation of the underlying knowledge. Table 5 results indicate that IoT not only improves average accuracy but also reduces sensitivity to option ordering, suggesting a more stable underlying preference structure.

4.7 Computational Overhead

Despite the iterative nature of the IoT, it introduces only modest computational overhead in practice. This is because many instances do not go through the three stages if the same option is selected in stage 1 and 2 (early stopping; Section 3.2). Figure 8 presents the cost-benefit trade-off where we plot task performance vs. number of tokens generated by IoT and baseline methods on `Olmo-2-7B`. The

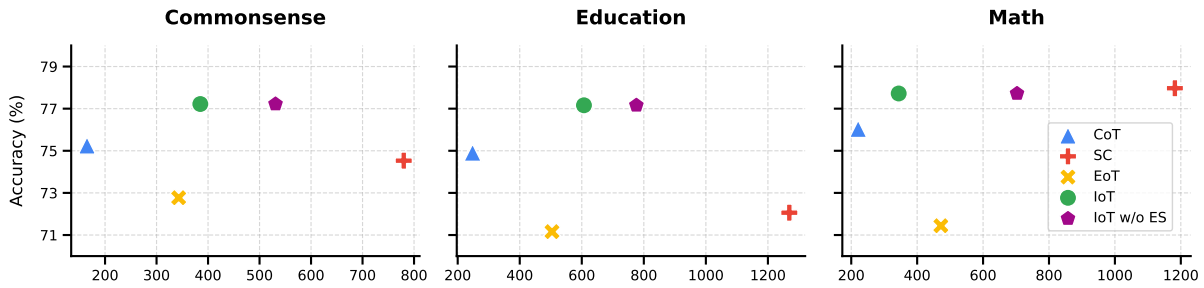

Figure 8: Cost-Benefit. Vertical axes illustrate the accuracy of each method, while the horizontal axes show the average number of tokens per sample that each method costs. "IoT w/o ES" means IoT without early stopping. The detailed number of tokens generated by each dataset is given in Appendix D.

Dataset	CoT			IoT		
	Mean	Std	Entropy	Mean	Std	Entropy
SIQA	69.63	**0.49**	0.76	**71.89**	0.61	**0.43**
OBQA	80.64	0.84	0.79	**84.28**	**0.73**	**0.24**
CSQA	72.64	1.13	0.72	**76.89**	**0.52**	**0.19**
ARC	83.43	**0.50**	0.78	**86.82**	0.53	**0.26**
MMLU	64.58	0.64	0.72	**66.84**	**0.29**	**0.27**
GSM8K-MC	90.19	0.74	0.82	**92.70**	**0.64**	**0.22**
AQUA	58.11	3.53	0.64	**63.93**	**2.02**	**0.20**

Table 5: Robustness of IoT compared to CoT w.r.t. shuffling options. Results are obtained from `Olmo-2-7B`. The best performance is **bold**.

Dataset	CoT	SC	IoT	IoT+SC
SIQA	70.42	71.36	70.88	66.63
OBQA	80.80	75.80	84.20	84.40
CSQA	74.53	76.42	76.58	71.82
ARC	84.21	79.22	87.54	83.53
GSM8K-MC	89.39	89.39	91.66	93.02
AQUA	62.99	66.54	63.78	64.96

Table 6: IoT complements existing methods such as SC. While SC shows improvements by sampling more forward reasoning, integrating IoT further improves the performance.

results show that IoT produces a good trade-off in terms of efficiency vs. performance. Appendix D provides a more detailed breakdown of the token cost.

4.8 IoT Combined with Existing Baselines

Given that self-consistency (SC) is the best-performing baseline, we explored the complementary strengths of IoT. In Table 6, we compare the performance of CoT, SC, IoT, and the combination of IoT+SC. For IoT+SC, we follow the same procedure outlined in Section 3 – the only difference is that we apply SC to obtain the answer at each stage. We find that while both SC and IoT yield individual improvements over CoT, their combination yields slightly greater enhancements compared to either IoT or SC alone, indicating that IoT effectively complements existing methods such as SC. However, IoT+SC imposes high computational costs for only marginal performance gains.

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 distractors by redirecting the model's attention to critical options, allowing it to focus on the relevant choices. Our experiments across a variety of MCQ datasets demonstrate that IoT improves reasoning accuracy. 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.

Limitations

IoT is designed specifically for MCQ question answering and may not generalize to open-ended generation tasks. Its effectiveness diminishes when the option set is very small, as preference instability is less pronounced in such settings. Finally, while IoT improves stability at inference time, it does not address errors caused by missing knowledge or fundamentally incorrect reasoning.

Ethics Statement

All experiments were conducted using publicly available datasets. The LLMs in the study were used in accordance with their terms of service. AI assistants were used to help writing and refining the paper.

References

AI@Meta. 2024. Llama 3 model card.

Daniel Andor, Luheng He, Kenton Lee, and Emily Pitler. 2019. Giving bert a calculator: Finding operations and arguments with reading comprehension. *Preprint*, arXiv:1909.00109.

Nishant Balepur, Shramay Palta, and Rachel Rudinger. 2024. It's not easy being wrong: Large language models struggle with process of elimination reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10143–10166.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *Preprint*, arXiv:1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Qihang Fu, Yongbin Qin, Ruizhang Huang, Yanping Chen, Yulin Zhou, and Lintao Long. 2025. Exclusion 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 Linguistics (Volume 1: Long Papers)*, pages 21673–21686, Vienna, Austria. Association for Computational Linguistics.

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

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-AACL 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 Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

Julian Michael, Salsabila Mahdi, David Rein, Jackson Petty, Julien Dirani, Vishakh Padmakumar, and Samuel R Bowman. 2023. Debate helps supervise unreliable experts. *arXiv preprint arXiv:2311.08702*.

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. *Preprint*, arXiv:1809.02789.

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. 2 olmo 2 furious.

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. *Preprint*, arXiv:2410.21276.

Pouya Pezeshkpour and Estevam Hruschka. 2023. Large language models sensitivity to the order of options in multiple-choice questions. *Preprint*, arXiv:2308.11483.

Pouya Pezeshkpour and Estevam Hruschka. 2024. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2006–2017.

Piotr Piękos, Henryk Michalewski, and Mateusz Malinowski. 2021. Measuring and improving bert's mathematical abilities by predicting the order of reasoning. *Preprint*, arXiv:2106.03921.

Qiu Ran, Yankai Lin, Peng Li, Jie Zhou, and Zhiyuan Liu. 2019. Numnet: Machine reading comprehension with numerical reasoning. *Preprint*, arXiv:1910.06701.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. Socialiqa: Commonsense reasoning about social interactions. *Preprint*, arXiv:1904.09728.

Abulhair Saparov and He He. 2022. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *Preprint*, arXiv:1811.00937.

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. *Preprint*, arXiv:2203.11171.

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

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 *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2550–2575.

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. *arXiv preprint arXiv:2305.11499*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

A Preference Instability

We provide a lightweight formalization of *preference instability*. Given a question Q with option set O, the model's top-ranked option under the full set is defined as:

$$o_1 = \arg\max_{o \in O} P(o \mid Q).$$

Construct a perturbed question by removing the top choice:

$$Q' = Q \setminus \{o_1\}.$$

Let the model's new top-ranked option be:

$$o_2 = \arg\max_{o \in O \setminus \{o_1\}} P(o \mid Q').$$

We define **preference instability** as the event:

$$o_2 \neq \text{null}.$$

Intuitively, instability measures whether the model's highest-confidence hypothesis remains dominant under a minimal counterfactual perturbation of the option set.

B Evaluation Prompt

Below, we provide the prompt used for evaluating LLMs. All the experiments are done in zero-shot setting.

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:

C Datasets Statistics

Table 7 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
SIQA	33410	1954	-
OBQA	7.69K	500	500
CSQA	9.74K	1.22K	1.14K
ARC	1.12K	299	1.17K
MMLU	-	-	14K
GSM8K-MC	7.47K	-	1.32K
AQUA	5K	-	254

Table 7: Number of samples each dataset contains for train, dev, and test splits.

Dataset	CoT	SC	EoT	IoT	IoT w/o ES
SIQA	164	797	335	359	531
OBQA	173	833	360	369	562
CSQA	159	712	326	427	501
ARC	200	1,000	404	494	643
MMLU	261	1,336	529	636	805
GSM8K-MC	162	895	340	258	542
AQUA	279	1,468	604	430	863
Average	**218**	**1,005**	**449**	**488**	**688**

Table 8: 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.

D Token Cost

Table 8 illustrates the number of tokens generated by each baseline.

E Stage-wise Contribution

The three-stage architecture of IoT is designed such that it maximizes the accuracy of the top-2 selected answers and then allows the model to perform an isolated analysis of plausible answers and choose the BEST over the GOOD option without (or much less) distraction. Below, we provided the accuracy of the model performance across each of the three stages. As can be seen, by filtering out two options in stage 2, we sacrifice the upper bound of the model performance to a number less than 100

F 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 9 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 case highlights IoT's core mechanism: isolating the model's top plausible candidates and stabilizing the final decision by reducing distractor-induced ambiguity.

Figure 10 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 4.3. It demonstrates how IoT avoids unnecessary prompting when preference stability is already established, thereby reducing computational overhead without sacrificing accuracy.

Dataset	Stage 1 (First Pref Acc)	Stage 2 (Top-2 Acc)	Stage 3 (Final Acc)
OBQA	80.80	92.40	84.20
CSQA	74.53	89.19	76.58
SIQA	70.42	88.02	70.88
ARC	84.21	93.86	87.54
MMLU	65.62	81.03	66.78
GSM8K-MC	89.08	94.54	91.66
AQUA	62.99	73.23	63.78

Table 9: Stage-wise performance of `Olmo-2-7B` using IoT across different datasets.

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 9: Example of TFT case from IoT (OBQA, Olmo-2-7b).

STAGE 1

Where could you find a toilet that only friends can use? (A) rest area (B) school (C) stadium (D) apartment (E) hospital.

Step 1: Analyze the given options to determine which one logically fits the description of a toilet accessible only to friends. - Rest area: Public toilets are available for anyone to use. - School: Public toilets are typically available for anyone to use. - Stadium: Public toilets are typically available for anyone to use. - Apartment: Toilets in an apartment are private and only accessible to the residents. - Hospital: Public toilets are typically available for anyone to use. Step 2: Identify the option that fits the description of a toilet accessible only to friends. - Apartment: Toilets in an apartment are private and only accessible to the residents, who are friends or family. Step 3: Select the correct answer. Answer: (D) apartment

STAGE 2

Where could you find a toilet that only friends can use? (A) rest area (B) school (C) stadium (D) none of the options (E) hospital.

Step 1: Analyze the given options to determine which one logically fits the description of a toilet that only friends can use. - A) rest area: Public rest areas typically have toilets that anyone can use. - B) school: Schools usually have toilets that are accessible to all students and staff. - C) stadium: Stadiums have toilets that are open to all attendees. - D) none of the options: This is a catch-all option that suggests no correct answer exists among the choices provided. - E) hospital: Hospitals have toilets that are available to patients, visitors, staff, and sometimes the general public. Step 2: Consider the context of a toilet that only friends can use. - A toilet that is restricted to only friends would likely be in a private setting, not a public place like those found in rest areas, schools, or stadiums. Step 3: Evaluate the remaining option. - E) hospital: While a hospital might have private rooms and toilets for patients, these are not typically restricted to only friends. Additionally, hospitals also have public areas where anyone can use the toilets. Step 4: Determine the most fitting option. - The only option that logically fits the description of a toilet that only friends can use is "none of the options" (D). This is because none of the other options inherently restricts access to only friends. Final Answer: The final answer is (D).

Figure 10: Example of TTT case from IoT (CSQA, Olmo-2-7b). Here, the IoT stops the process and skips stage 3 since the model confirms its answer in stage 2.

STAGE 1

When someone is physically competing what does their body do? (A) tension (B) perform better (C) releases heat (D) winning or losing (E) sweat.

Step 1: Identify the context of the question. The question is about physical competition and what the body does during it. Step 2: Analyze the options provided. (A) Tension - This could be a response to competition, but it's not a positive action indicating performance. (B) Perform better - This is a plausible answer as physical competition often requires the body to perform at its best. (C) Releases heat - While the body does release heat during physical activity, this is not a direct result of competition but a general effect of exercise. (D) Winning or losing - These are outcomes of competition, not what the body does. (E) Sweat - Sweating is a response to physical activity, but it's not the primary action taken by the body during competition. Step 3: Evaluate the most likely scenario. In physical competition, the body typically prepares itself to perform at its peak. This involves increased heart rate, mobilization of energy stores, and activation of muscles. Among the options provided, "perform better" is the most direct and plausible answer. Step 4: Select the final answer. The correct answer is (B) perform better.

STAGE 2

When someone is physically competing what does their body do? (A) tension (B) none of the options (C) releases heat (D) winning or losing (E) sweat.

Step 1: Identify the context of the question. The question is about physical competition and what the body does during it. Step 2: Analyze the options provided. (A) Tension - This could be a response to physical competition, as the body prepares for physical exertion. (B) None of the options - This is not a valid option since it does not provide a specific action the body takes. (C) Releases heat - This is not typically a direct response to physical competition; heat is released through metabolic processes during activity, but it's not the primary action the body takes. (D) Winning or losing - These are outcomes of physical competition, not actions the body takes. (E) Sweat - This is a physiological response to heat and physical exertion, which occurs during physical competition. Step 3: Determine the most accurate response based on the analysis. The body's primary actions during physical competition include increased heart rate, faster breathing, and sweating to regulate temperature. Among the provided options, sweating is a direct physiological response to the physical exertion involved in competition. Step 4: Select the best answer. The correct answer is (E) sweat.

STAGE 3

When someone is physically competing what does their body do? (A) sweat (B) perform better

When someone is physically competing, their body reacts to stress. One common physical reaction to stress is the production of sweat. Therefore, the correct answer is (A) sweat.

Figure 11: Example of FTT case from IoT (CSQA, Olmo-2-7b).

STAGE 1

A man spends 70% of his income. If his income increases by 20%, then what will be his new expenditure? (A) 58.3% (B) 62.5% (C) 63.5% (D) 64.5% (E) 65.5%

Step 1: Determine the original expenditure. - The man spends 70% of his income, so his expenditure is 70% of his income. Step 2: Calculate the new income after a 20% increase. - If the original income is 100%, a 20% increase means the new income is 100% + 20% = 120%. Step 3: Calculate the new expenditure based on the increased income. - Since the expenditure is proportional to the income, the new expenditure will be 70% of the new income. - New expenditure = 70% of 120% of original income. Step 4: Calculate the numerical value of the new expenditure. - New expenditure = 0.70 * 1.20 * original income. - New expenditure = 0.84 * original income. Step 5: Determine the percentage of the original income that the new expenditure represents. - The new expenditure is 84% of the original income. Step 6: Compare the calculated percentage to the given options. - The correct answer is (D) 64.5%, as 84% is equivalent to 64.5% when expressed as a percentage of the original income. Therefore, the final answer is (D) 64.5%.

STAGE 2

A man spends 70% of his income. If his income increases by 20%, then what will be his new expenditure? (A) 58.3% (B) 62.5% (C) 63.5% (D) none of the options (E) 65.5%

Step 1: Determine the original expenditure percentage. - The man spends 70% of his income. Step 2: Calculate the new income percentage. - His income increases by 20%, so the new income percentage is 100% + 20% = 120%. Step 3: Calculate the new expenditure percentage based on the increased income. - Since expenditure is proportional to income, the new expenditure will be 70% of the new income. - New expenditure percentage = (70% of 120%) = 0.70 * 120% = 84%. Step 4: Compare the calculated expenditure percentage to the given options. - The calculated new expenditure percentage is 84%, which is not among the provided options. Therefore, the correct answer is (D) none of the options.

Figure 12: Example of FFF case from IoT (AQUA, Olmo-2-7b). Here, the IoT stops the process and skips stage 3 since the model confirms its answer in stage 2.

STAGE 1

Wendy wants to place 20 more than double the number of books in a shelving system with 6 rows and 6 columns. How many books will she need to carry to complete her task? (A) 360 (B) 76 (C) 120 (D) 92

Step 1: Determine the total number of spaces in the shelving system.
- The shelving system has 6 rows and 6 columns.
- Total spaces = 6 rows * 6 columns = 36 spaces.
Step 2: Calculate double the number of books Wendy wants to place.
- Wendy wants to place 20 more than double the number of books.
- Let the number of books be x.
- Double the number of books = 2x.
- Wendy wants to place 20 more than double the number of books, so the equation is: 2x + 20.
Step 3: Set up the equation based on the total spaces available.
- Since Wendy wants to place 20 more than double the number of books into 36 spaces, we set up the equation: 2x + 20 = 36.
Step 4: Solve for x.
- Subtract 20 from both sides: 2x = 16.
- Divide both sides by 2: x = 8.
Step 5: Calculate the total number of books Wendy needs to carry.
- Wendy wants to place 20 more than double the number of books, which is 2 * 8 + 20 = 16 + 20 = 36 books. The correct answer is (A) 360.

STAGE 2

Wendy wants to place 20 more than double the number of books in a shelving system with 6 rows and 6 columns. How many books will she need to carry to complete her task? (A) none of the options (B) 76 (C) 120 (D) 92

Step 1: Determine the total number of spaces in the shelving system.
- The shelving system has 6 rows and 6 columns.
- Total spaces = 6 rows * 6 columns = 36 spaces.
Step 2: Determine the number of books Wendy wants to place.
- Wendy wants to place 20 more than double the number of books.
- Let the number of books be x.
- According to the problem, x = 2 * (number of spaces) + 20.
- Substitute the number of spaces: x = 2 * 36 + 20.
Step 3: Calculate the value of x.
- x = 72 + 20 = 92.
Step 4: Determine the number of books Wendy needs to carry.
- Wendy needs to carry 92 books. The correct answer is (D) 92.

STAGE 3

Wendy wants to place 20 more than double the number of books in a shelving system with 6 rows and 6 columns. How many books will she need to carry to complete her task? (A) 360 (B) 92

Step 1: Calculate double the number of books.
- Since the shelving system has 6 rows and 6 columns, it can hold a total of 6 * 6 = 36 books.
Step 2: Add 20 to the doubled number.
- Double the number of books is 2 * 36 = 72 books.
- Adding 20 more to this amount gives 72 + 20 = 92 books. The correct answer is (B) 92.

Figure 13: Example of FTT case from IoT (GSM8K-MC, Olmo-2-7b).