Project Report: Evaluating Logical Reasoning in Large Language Models

AKASH MITTAL

December 2025

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in natural language understanding and generation of text. However, their ability to perform strict logical reasoning—and thinking instead of probabilistically pattern matching is still a subject of intense scrutiny. This project investigates the robustness of LLMs when faced with formal logical puzzles, with different prompting strategies and emotional tones, and specific logical structures influencing model performance.

2 Research Questions

This study tries to answer the following core questions:

1. **Model Reliability:** Do LLMs truly understand logical constraints (e.g., negation, exclusion, quantifiers, order) or do they just rely on surface-level associations?
2. **Prompt Sensitivity:** How do the different prompting strategies (Zero-Shot, Few-Shot and Chain-of-Thought) and emotional tones (Neutral, Urgent, High-Stakes) impact accuracy?
3. **The "Urgency Trap":** Does applying time pressure (via urgent prompts) cause models to trade accuracy for speed, and if able to effectively "rush" their cognitive process?
4. **Reasoning vs. Intuition:** Does asking a model to "think step-by-step" (Chain-of-Thought) actually improves the performance compared to the ask for a direct answer?

3 Dataset Overview

- **Source:** `Puzzles_List_6_Each.csv` selected 6 puzzles for each category and difficulty level, randomly from main dataset at huggingface - `https://huggingface.co/datasets/datatune/LogiQA2.0`
- **Content:** A selected set of logical puzzles categorized by 15 different reasoning types as follows: -

Categorical Reasoning
Sufficient Conditional Reasoning
Necessary Conditional Reasoning
Conjunctive Reasoning
Categorical Reasoning, Sufficient Conditional Reasoning
Categorical Reasoning, Necessary Conditional Reasoning
Categorical Reasoning, Conjunctive Reasoning
Necessary Conditional Reasoning, Sufficient Conditional Reasoning
Conjunctive Reasoning, Sufficient Conditional Reasoning
Conjunctive Reasoning, Necessary Conditional Reasoning
Categorical Reasoning, Necessary Conditional Reasoning, Sufficient Conditional Reasoning
Categorical Reasoning, Conjunctive Reasoning, Sufficient Conditional Reasoning
Categorical Reasoning, Conjunctive Reasoning, Necessary Conditional Reasoning
Conjunctive Reasoning, Necessary Conditional Reasoning, Sufficient Conditional Reasoning
Categorical Reasoning, Conjunctive Reasoning, Necessary Conditional Reasoning,...
....Sufficient Conditional Reasoning

Structure:

- **Puzzle Text:** The scenario describing the constraints for each puzzle type.
- **Question:** The specific logical query.
- **Options:** Multiple-choice answers (0,1,2,3).

4 Models Used

Two distinct models were selected to represent different scales of capability and deployment: *Note: Both the models have been used though "ollama" for ease of running the model on laptop.*

Gemma 3:1b (Google DeepMind)

- *Type:* Ultra-lightweight, local model.
- *Role:* Represents edge-device capabilities. This model is expected to struggle with complex reasoning but it offers high speed and use. Gemma 3 models are multimodal, handling text and image input and generating text output, with open weights for both pre-trained variants and instruction-tuned variants. Gemma 3 has a large, 128K context window. Gemma 3 models are well-suited for a variety of text generation and image understanding tasks, including question answering, summarization, and reasoning. Being a small model, it was easy to deploy it on laptop.

GPT-OSS:20b-cloud

- *Type:* Mid-to-large scale open-source model (cloud-hosted).
- *Role:* Represents a robust server-side model. Expected to handle deeper reasoning tasks and instruction following more effectively. OpenAI gpt-oss 20B is a 20B open-weight language model released under the Apache 2.0 license. It is well-suited for reasoning and function calling use cases. The model is optimized for deployment on consumer hardware. The 20B model delivers similar results to OpenAI o3-mini on common benchmarks and can run on edge devices with 16GB of memory, making it ideal for on-device use cases, local inference, or rapid iteration without costly infrastructure.

5 Methodology

The evaluation pipeline consisted of three different analytical phases:

5.1 Phase A: The Prompt Grid Search

I tested every puzzle against a 3x3 matrix of prompt configurations (for prompt tones and prompt types):

- **Strategies:** Zero-Shot (ZS), Few-Shot (FS), Chain-of-Thought (CoT).
- **Tones:** Neutral, Urgent ("Respond quickly!"), High Stakes ("Critical decision!").
- **Metric**: Accuracy and Response Time (Seconds).

5.2 Phase B: Logical Gap Analysis

I tagged every puzzle with specific logic keywords (e.g., "not", "except", "all") to measure:

- **Performance Drop:** The difference in accuracy between puzzles *with* these terms vs. puzzles *without* them.

5.3 Phase C: Counterfactual Ablation ("Sanity Check")

I removed critical logic words (e.g., deleting "not" from "is not") from the puzzle text and re-ran the models, to challenge the model and evaluate if the model actually is paying attention or not.

- **Sensitivity Score:** The percentage of times the model changed its answer.
- **Interpretation:** When the model keeps the same answer after "not" is removed, it means it was never reading the logic in the first place (thus, it was just doing a guesswork, probabilistically).

6 Analysis & Visualization Results

6.1 Performance Matrix (Accuracy by Strategy & Tone)

How did the models perform across different prompting conditions?

Table 1: Performance Matrix across different configurations

Model	Tone	Zero-Shot (ZS)	Few-Shot (FS)	Chain-of-Thought (CoT)
Gemma 3:1b	Neutral	**34.7%**	21.3%	22.7%
	Urgent	29.3%	22.7%	29.3%
	High Stakes	33.3%	24.0%	29.3%
GPT-OSS:20b	Neutral	**81.3%**	81.3%	**81.3%**
	Urgent	80.0%	80.0%	76.0%
	High Stakes	77.3%	76.0%	78.7%

- **Gemma 3:1b:** Best performance was achieved with simple **Zero-Shot Neutral** prompts. Complex prompting (CoT, FS) degraded performance, likely due to context overload.
- **GPT-OSS:** Highly stable across the matrix. **High Stakes** prompts consistently *lowered* accuracy compared to Neutral tone, indicating a "performance anxiety" phenomenon where added pressure induced *hallucinations. Further, accuracy almost all the prompts types and tones is around 80%*

6.2 The "Urgency Trap" (Time vs. Accuracy)

Did "Urgent" prompts actually make the models faster or just created a fuss?

- **Findings:** Gemma 3:1b showed a clear **Speed-Accuracy Trade-off**. It took the instruction "respond quickly and urgently" very literally, thus reducing the computation time by $\tilde{2}0\%$ but at the same time failing significantly more often.

Table 2: Time Analysis comparing Neutral and Urgent tones

Model	Neutral (s)	Urgent (s)	Speedup (%)	Accuracy Impact
Gemma 3:1b	1.56s	1.24s	**20.5% Faster**	**-5.4% Accuracy**
GPT-OSS	11.33s	10.42s	8.0% Faster	-1.3% Accuracy

6.3 Logic Blindness (Performance Drop by Logic Type)

Which logical concepts confused the models the most?

Gap Definition. The performance gap is defined as " $\mathbf{Acc}_{\mathbf{Without\ Term}} - \mathbf{Acc}_{\mathbf{With\ Term}}$ " Positive values indicate that the presence of a logic term degrades model performance, while negative values indicate performance improvements when the term is explicitly stated.

Table 3: Logic Sensitivity Analysis

Model	Logic Category	Acc (With)	Acc (Without)	Gap
GPT-OSS	Exclusion	74.60%	80.15%	**+5.54**%
GPT-OSS	Quantifiers	78.55%	79.86%	+1.31%
GPT-OSS	Negation	81.13%	74.24%	**-6.89**%
GPT-OSS	Ordering	87.58%	76.63%	**-10.95**%
Gemma	Exclusion	30.16%	26.78%	-3.38%
Gemma	Quantifiers	28.94%	25.35%	-3.59%
Gemma	Negation	27.67%	26.77%	-0.91%
Gemma	Ordering	26.80%	27.59%	+0.79%

Category-Level Interpretation As shown above in Table 3, the category-level logic constructs induce relatively small performance gaps once results are averaged across prompt variants. For "*GPT-OSS*", "***Exclusion***" produces a modest degradation of **5.54**%, while ***Quantifiers*** result only in ***1.31***% drop. Further, Ordering and Negation yield negative gaps (*-10.95*% and *-6.89*%, respectively), indicating improved performance when these logical relations are explicitly stated in the prompts.

For "*Gemma*", the gaps across all categories remain within ±4%, indicating no strong systematic sensitivity to abstract logic categories. This indicates the failure in gemma model in overall rather than specific to any logical types.

Table 4: Word-Level Keyword Sensitivity

Model	Keyword	Acc (With)	Acc (Without)	Gap
GPT-OSS	without	60.32%	81.05%	**+20.73**%
GPT-OSS	some	65.28%	80.76%	**+15.49**%
GPT-OSS	all	76.67%	80.74%	+4.07%
GPT-OSS	only	77.16%	79.73%	+2.57%
GPT-OSS	not	78.68%	79.53%	+0.85%
Gemma	except	24.44%	27.62%	+3.17%
Gemma	only	25.31%	28.07%	+2.76%
Gemma	after	25.93%	27.54%	+1.61%
Gemma	last	25.93%	27.54%	+1.61%
Gemma	first	26.67%	27.46%	+0.79%

Word-Level Interpretation As shown in Table 4 above, we see substantially larger gaps at the lexical level. For *GPT-OSS*, the presence of the keyword "without" leads to a 20.73 percentage point accuracy

drop, while "some" induced a 15.49 point degradation. The large positive gaps indicate higher sensitivity to keywords that introduce implicit negation or scope ambiguity.

However, Gemma shows smaller but consistent degradations for exclusion-related terms such as "except" (+3.17%) and "only" (+2.76%). Although overall accuracy remains low, the absence of extreme gaps suggests diffuse weakness rather than keyword-triggered failures.

6.4 Behavioral Analysis

If Model Noticed the removal of constraints?

Table 5: Logic Sensitivity Scores (% of answers changed after removing logic terms).

Model	Exclusion	Negation	Ordering	Quantifiers
Gemma	35.71%	13.21%	18.75%	6.98%
GPT-OSS	57.14%	20.75%	18.75%	20.93%

Logic Sensitivity Analysis. The data in Table 5 shows logic sensitivity scores, which is defined as the percentage of answers that change when logic-specific keywords are removed. Higher values indicate that model's reasoning behavior depends on the presence of the logic term, while low values indicate logic blindness. Both models exhibit high sensitivity to Exclusion, with GPT-OSS changing its answers in 57.14% of cases and Gemma in 35.71%. However, this behavioral sensitivity does not translate into improved accuracy, suggesting that models detect exclusion cues but frequently apply them incorrectly. In contrast, Gemma shows near-blindness to Quantifiers (6.98%), indicating that such constraints are often ignored entirely.

Combined with accuracy gap analysis, these results distinguish between two failure modes: (1) logic blindness, where the model fails to register a constraint, and (2) logic misuse, where the model reacts to the constraint but applies it incorrectly. Overall, accuracy gaps and sensitivity scores provide complementary evidence: word-level logic terms trigger strong behavioral reactions, yet these reactions often fail to yield correct reasoning. This suggests that current models possess surface-level awareness of logical cues without reliable compositional understanding.

6.5 Chain-of-Thought (CoT) Divergence

Did reasoning step-by-step help?

For **GPT-OSS**, while the *net accuracy* between Zero-Shot and CoT was identical (81.3%), the underlying behavior was different:

- **Disagreement Rate:** 10.7% (The models gave different answers for 8 out of 75 puzzles).
- **The "Overthinking" Effect:** In 4 cases, the Zero-Shot model was correct, but the CoT model "talked itself" into a wrong answer.
- **The "Reasoning Payoff":** In 4 other cases, reasoning successfully solved a puzzle that Zero-Shot missed.

The results show that CoT reasoning does not always improve performance. For GPT-OSS, it provided a behavioral shift without net accuracy gain, but for Gemma, the reasoning often hurts, thus highlighting the fragility of step-by-step inference in smaller models.

7 Interpretation of Results

7.1 Complexity Penalty

Smaller models such as **Gemma (1B)** demonstrated severe limitations when exposed to complex prompting strategies such as Few-Shot or Chain-of-Thought (CoT). The results show that CoT reduces overall accuracy

by ~12% for Gemma as compared to neutral prompts, indicating that the model lacks the attention span and representational capacity to process additional context effectively.

- Extra context, intended to guide reasoning, often acts as *noise*, confusing the model rather than helping.
- Larger models (GPT-OSS 20B) are more stable under complex prompting, though CoT shows negligible net gain (0%).
- **Recommendation:** For small models, simpler, direct prompts are safer and more effective.

7.2 The "Exclusion" Blind Spot

Both models exhibit a systematic weakness when reasoning about **negative constraints** such as "except," or "without,".

- **Accuracy (final aggregated, Category-Level):** GPT-OSS scores 74.6% on Exclusion puzzles (with the keyword) versus 80.2% when the keyword is removed, giving a modest **gap of +5.5%**.
- **Sensitivity:** Despite the small accuracy gap, GPT-OSS reacts to 57% of puzzles when Exclusion terms are removed, indicating the model *notices* the term, but the response does not substantially improve accuracy.
- **Interpretation:** The model relies on *surface cues* rather than fully understanding the logical structure implied by exclusion, often misapplying or ignoring the negative constraint.
- **Practical Implication:** Tasks involving negative constraints remain challenging for current LLMs; and the explicit logical checks or structured templates may be necessary.

7.3 Urgency is Dangerous

Prompts emphasizing urgency (e.g., "Important!", "Urgent" or "High-Stakes!") can harm reasoning performance, especially for smaller models:

- **Gemma:** Accuracy decreases under urgency-instructed prompts, likely due to attempting faster responses at the expense of logical rigor.
- **GPT-OSS:** Speed gains of 20.5%, but occasional instability in reasoning is observed.
- **Recommendation:** LLMs do not benefit from human-like pressure cues, so one can avoid mentionaing urgency in prompts.

7.4 Reasoning is a Double-Edged Sword (Chain-of-Thought Divergence)

Step-by-step reasoning (CoT) does not universally improve performance. The analysis reveals nuanced behavior:

- **GPT-OSS:** Net accuracy unchanged (0% CoT gain), but 10.7% of answers differed between Zero-Shot (ZS) and CoT. Of these:
  - 4 cases corrected previous errors (*Reasoning Payoff*)
  - 4 cases introduced new errors (*Overthinking Effect*)
- **Gemma:** CoT reduced overall accuracy by 12%, indicating that step-by-step reasoning can mislead smaller models, often overcomplicating simple inference tasks.
- **Interpretation:** CoT benefits depend on model capacity. Large models can redistribute correct answers without net gain, while smaller models are prone to "hallucination loops," producing plausible but incorrect justifications.

7.5 Overall Insights

Across all analyses, we observe that:

- High **sensitivity scores** indicate that models notice critical logic keywords.
- Modest **accuracy gains** or even losses reveal that detection does not guarantee correct application of logical reasoning.
- Smaller models are particularly vulnerable to *surface-level heuristics*, prompt complexity, and CoT misapplications.
- Large models show stronger stability but still exhibit systematic weaknesses on negative constraints (Exclusion) and occasionally "overthink" when reasoning step-by-step.

These findings emphasize that LLMs possess *surface-level awareness* of logical cues without consistently achieving *compositional reasoning understanding*.

8 Conclusion

This study highlights that *prompt engineering must be model-specific*. Larger models like GPT-OSS exhibit relative robustness but still retain systematic "blind spots," particularly on puzzles involving **exclusion logic**. Smaller models such as Gemma 3:1b illustrate that for low-capacity LLMs, *simpler, neutral prompts can outperform complex or step-by-step strategies*, highlighting the principle that "**less is more**."

For future applications relying on LLMs for logical decision-making, it is crucial to implement **verification layers** for critical logical constructs, especially **exclusion** and **negation**, which remain the most fragile aspects of current model reasoning.

Note: This analysis was conducted on a limited subset of puzzles (5 per category across 15 categories), which may introduce sampling bias. A larger, more diverse dataset is required to validate and generalize these findings.

9 Plots

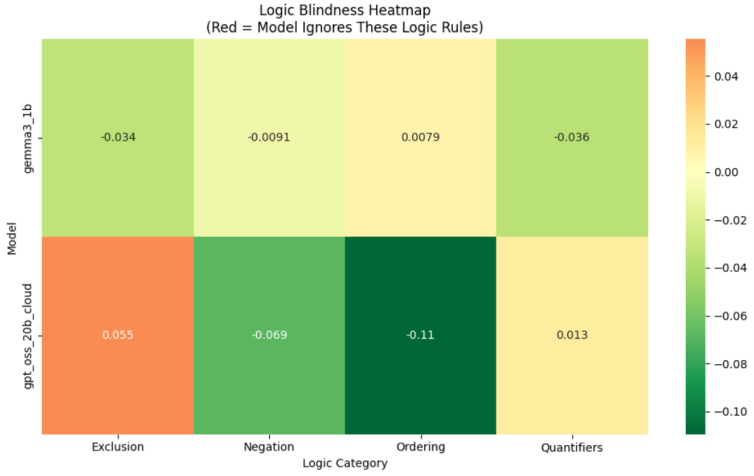

Figure 1: Logic Blindness Heatmap

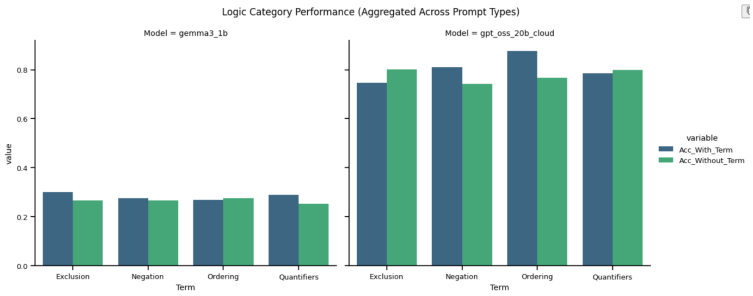

Figure 2: Performance Drop by Logic Terms

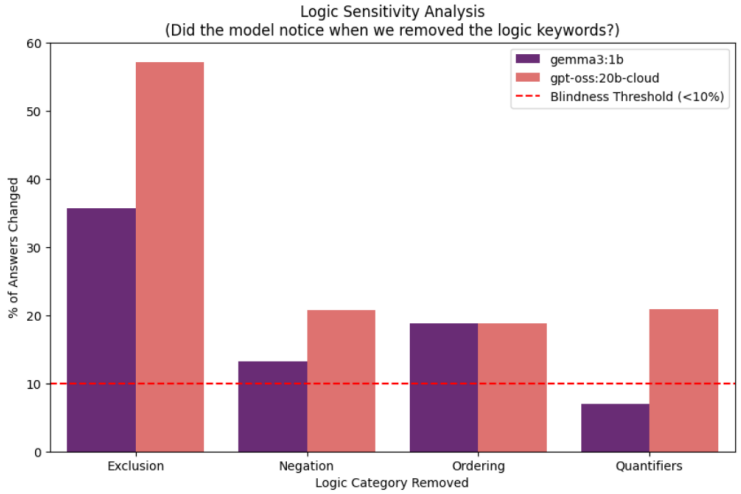

Figure 3: Logic Sensitivity Analysis

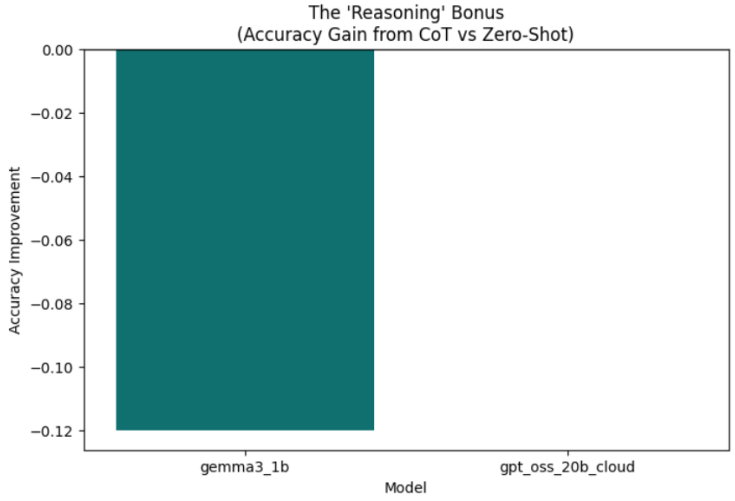

Figure 4: Reasoning Bonus

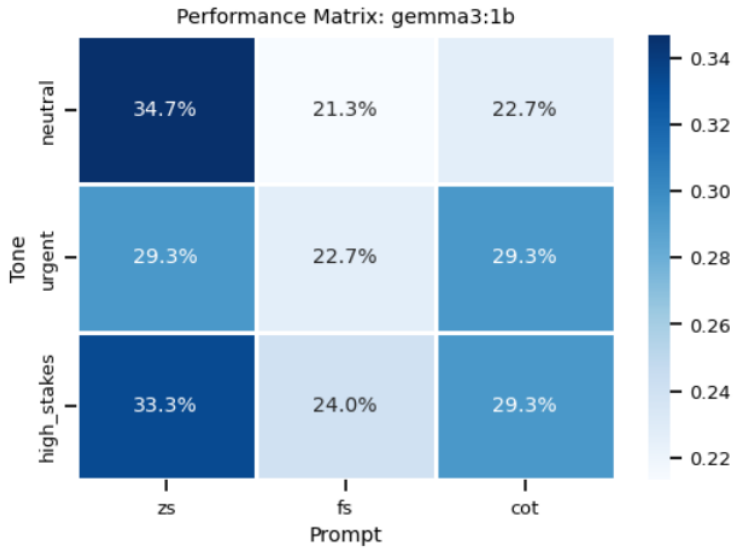

Figure 5: Gemini Performance Matrix

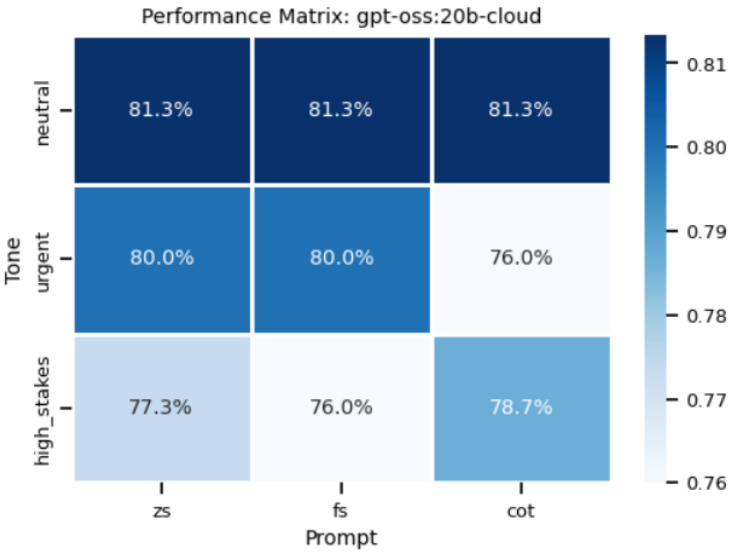

Figure 6: GPT Performance Matrix

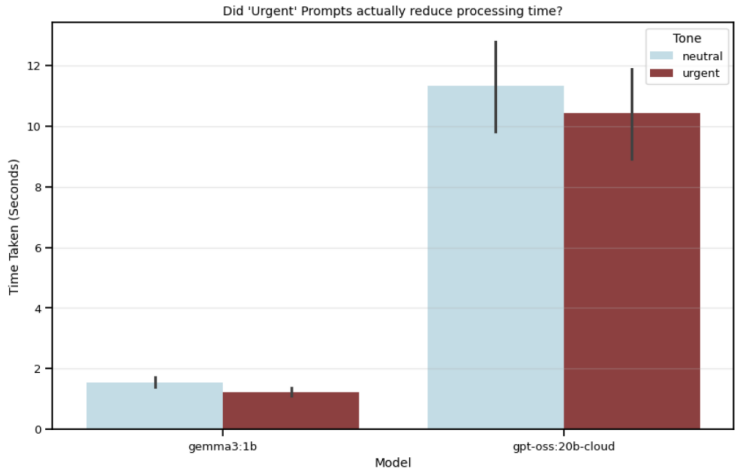

Figure 7: Time Analysis (Urgent vs Neutral) for Both Models

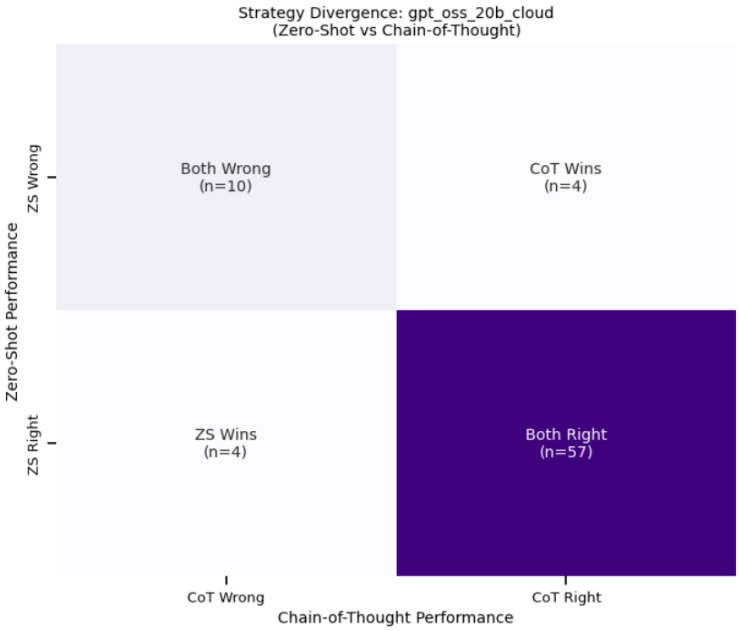

Figure 8: Strategy Divergence

Disclaimer:

AI Usage: *I have used Gemini 3 to structure the report and get help with LaTeX formats. Further, I used it for grammatical errors and for comprehensiveness or report generation and a partial usage was for code generation for certain simplified code chunks. However, the reasoning and approach followed throughout the project is individually mine, and I take full responsibility for correctness of the results.*

References

LogicQA. LogiQA is constructed from logical comprehension problems from publicly available questions of the National Civil Servants Examination of China, designed to test candidates' critical thinking and problem-solving abilities.

Tang, X., Zheng, Z., Li, J., Meng, F., Zhu, S. C., Liang, Y., & Zhang, M. (2023). *Large language models are in-context semantic reasoners rather than symbolic reasoners.* arXiv preprint arXiv:2305.14825.

Sullivan, R., & Elsayed, N. (2024). *Can Large Language Models Act as Symbolic Reasoners?* arXiv preprint arXiv:2410.21490.