Final Project

Jeongwon Bae (945397461)

Dec. 6, 2024

1 Introduction

1.1 Background and Motivation

The recent advancements of Large Language Models (LLMs) such as GPT-4o [1] and Claude 3.5 Haiku [2] has significantly transformed the field of natural language processing. These models have demonstrated remarkable abilities in tasks like language translation, text summarization, and conversational dialogue generation. However, as their use becomes more widespread, concerns about their reliability and trustworthiness have grown, especially in high-stakes applications such as healthcare, legal advice, and education [3]. A critical aspect that remains underexplored is the metacognitive abilities of these modelstheir capacity to recognize and reflect upon their own knowledge and reasoning processes. In human cognition, metacognition plays a vital role in learning, problem-solving, and decision-making [4]. It involves awareness of one's cognitive processes, including the ability to detect errors, evaluate the certainty of one's knowledge, and adjust strategies accordingly.

For LLMs, developing metacognitive abilities could enhance their performance by enabling them to detect and correct errors in their outputs, assess the certainty of their responses, and maintain consistency across interactions [5][6]. These capabilities are essential for building user trust and ensuring that LLMs provide accurate and reliable information. One area where metacognitive abilities are particularly important is in handling faulty questions that contain logical inconsistencies, incorrect premises, or impossible conditions. An LLM's ability to recognize such faults, acknowledge them, and respond appropriately is crucial for preventing the propagation of misinformation and for assisting users in understanding and correcting misunderstandings.

1.2 Objectives and Research Questions

The primary objective of this research is to evaluate the metacognitive abilities of current LLMs by focusing on three key competencies: error detection ability, error admission and correction ability, and context maintenance ability. Specifically, we aim to investigate whether LLMs can recognize and explicitly point out errors in faulty questions, acknowledge their own incorrect answers when errors are pointed out and subsequently provide correct responses, and maintain consistency over repeated interactions by adjusting their responses based on previous exchanges.

To address these objectives, we formulate the following research questions:

1.2.1 Error Detection in Faulty Questions

- How effectively can LLMs detect errors in questions that contain logical inconsistencies, incorrect premises, or impossible conditions?
- What types of errors are more likely to be detected or missed by different models?

1.2.2 Error Admission and Correction After Feedback

- When provided with feedback pointing out an error in their initial response, how do LLMs react?
  - Do they admit the error and provide a corrected answer?
  - Do they justify their initial response without acknowledging the mistake?
- Is there a difference in error admission and correction behaviors among different LLMs?

1.2.3 Consistency and Adaptation Over Repeated Interactions

- How consistent are LLMs in their responses when the same faulty question is asked again?
- Can LLMs recognize and correct their initial mistakes without external feedback when questions are repeated?

By exploring these questions, we aim to uncover the strengths and limitations of current LLMs in terms of metacognitive processing. Understanding these aspects is essential for improving model training methodologies to enhance error detection and correction capabilities, developing better interaction protocols that encourage LLMs to reflect on their responses, and informing users about the reliability of LLM-generated information.

1.3 Research Contributions and Report Overview

This study contributes to the field in several ways. Theoretically, it provides insights into the metacognitive abilities of LLMs, a relatively unexplored area in artificial intelligence research. Practically, it offers guidance for developers and practitioners on how to improve LLM performance, particularly in applications where accuracy and reliability are critical. Additionally, it establishes evaluation metrics and experimental protocols that can be used for future assessments of LLM metacognition.

The remainder of this report is organized as follows. First, we review existing literature on LLMs' error handling and metacognitive capabilities. Next, we detail the experimental design, including the selection of LLMs, construction of faulty question sets, and evaluation metrics. We then present the findings from the experiments, including quantitative metrics and qualitative analyses. Following that, we interpret the results, discuss their implications, and explore potential reasons for observed behaviors. Finally, we summarize the key findings, acknowledge limitations, and suggest directions for future research. This research sheds light on the metacognitive capacities of modern LLMs and provide actionable insights for enhancing their reliability and effectiveness in real-world applications.

2 Related Work

2.1 LLMs' Error Handling Capabilities

The rapid advancement of LLMs has significantly improved natural language processing tasks. However, several studies have highlighted limitations in their ability to detect and handle errors effectively. One common issue is that LLMs often fail to recognize inconsistencies or factual inaccuracies in the input text, leading to the generation of incorrect or misleading responses. For instance, models like GPT-3 have been shown to produce plausible-sounding but incorrect answers, a phenomenon sometimes referred to as "hallucination" [7].

Research has documented cases where LLMs provide confident answers to unanswerable or nonsensical questions, raising concerns about their reliability in real-world applications [8]. Even when presented with subtly flawed questions, LLMs may generate responses without detecting the embedded errors. This limitation can result in the propagation of misinformation, especially in high-stakes domains such as healthcare and legal advice.

Moreover, even when LLMs recognize that there is an inconsistency or error in the input, they may not respond appropriately. Instead of expressing uncertainty or seeking clarification, they might generate vague or evasive answers. Zhang et al. observed that some models tend to produce non-committal responses when faced with ambiguous queries, without explicitly addressing the underlying issues [9].

Efforts to improve error detection and handling in LLMs have included training models on datasets containing unanswerable questions or adversarial examples. For example, the SQuAD 2.0 dataset introduced unanswerable questions to encourage models to recognize when a question cannot be answered based on the provided context [10]. Techniques such as uncertainty estimation and confidence scoring have also been explored to enable models to express doubt when necessary [11]. Despite these advances, there remains a significant gap in equipping LLMs with robust mechanisms for error detection and appropriate response generation.

2.2 Metacognition and Artificial Intelligence

Metacognition refers to the awareness and regulation of one's own cognitive processes, including the abilities to monitor understanding, detect errors, and adjust strategies accordingly. In human learning, metacognition plays a crucial role in effective problem-solving and decision-making, enabling individuals to recognize when they do not know something and to take steps to acquire the necessary knowledge. Flavell first introduced the concept, emphasizing its importance in cognitive development [4].

Applying metacognitive principles to artificial intelligence involves equipping AI systems with the capacity to assess their own performance and adapt based on that assessment. In the context of LLMs, metacognitive abilities could enable models to detect inconsistencies or errors in their inputs or outputs, express uncertainty appropriately, and adjust their responses in light of new information.

The applicability and necessity of metacognitive abilities in AI have been increasingly recognized, particularly as AI systems are deployed in domains where errors can have significant consequences. By incorporating metacognitive strategies, LLMs can become more reliable and trustworthy, enhancing user experience and safety. For instance, an LLM with metacognitive capabilities might recognize when a question contains a logical fallacy and inform the user, rather than providing an incorrect answer.

Recent research has begun to explore methods for integrating metacognitive functions into AI systems. Cox and Raja discuss the concept of metareasoning, where AI systems monitor and regulate their reasoning processes to improve performance [12]. Approaches include developing architectures that allow for self-monitoring and reflection, as well as training models to recognize and respond to their own limitations. The role of metacognition in improving the transparency and interpretability of AI systems has also been highlighted, as it enables models to provide explanations for their decisions and acknowledge uncertainty [13].

Incorporating metacognitive abilities into LLMs is essential for enhancing their reliability and aligning them more closely with human cognitive processes. It can lead to AI systems that not only perform tasks effectively but also understand and communicate their own limitations, thereby building greater trust with users.

3 Methodology

3.1 Selection of Large Language Models

To evaluate the metacognitive abilities of current LLMs, we selected five state-of-the-art models that represent a diverse range of architectures and capabilities. The models were chosen based on their performance, accessibility, and compatibility with available computational resources, ensuring a balance between diversity and efficiency. The selected models are:

- GPT-4o (Released on August 6, 2024)
- Claude 3.5 Haiku (Released on October 22, 2024)
- Llama 3.2-3B-Instruct
- Qwen 2.5-3B-Instruct
- Gemma-2-2B-It

3.2 Model Descriptions

GPT-4o is the latest iteration in OpenAI's GPT series, offering significant advancements in natural language understanding and generation. With an extended context window of up to 128,000 tokens and support for multimodal inputs (text, image, and audio), GPT-4o provides enhanced capabilities over its predecessors [1]. It is designed for high-volume API usage and includes features such as internet access for up-to-date information.

Claude 3.5 Haiku is a compact and efficient model from Anthropic's Claude series. Known for its near-instantaneous response times, it features a 200,000-token context window and excels in generating human-like outputs [2]. The model is optimized for automating basic repetitive tasks and is cost-effective, making it suitable for a wide range of applications.

Llama 3.2-3B-Instruct is a lightweight model from Meta AI's Llama family, focused on on-device inference and edge computing applications [14]. With 3 billion parameters, it supports BFloat16 numerics and is compatible with Android and iOS devices. The model offers tool-calling capabilities and is optimized for efficient deployment across various devices.

Qwen 2.5-3B-Instruct is part of the Qwen series, ranging from 0.5B to 72B parameters. The 3B-Instruct model incorporates both pretraining and post-training stages, utilizing a transformer architecture with advanced features like Rotatory Position Embedding and Grouped Query Attention [15]. It supports a full context length of 32,768 tokens and is designed for instruction following, long text generation, and chatbot applications.

Gemma-2-2B-It is a compact and efficient model from Google's Gemma family [16]. With 2 billion parameters, it is a decoder-only transformer designed for text-to-text generation tasks. Built upon the same research and technologies used to create the Gemini models, Gemma-2-2B-It supports English and is available in both pre-trained and instruction-tuned variants with open-source weights. Trained on a diverse dataset of over 2 trillion tokens, including web documents, code, and mathematical texts, it excels in tasks such as question answering, summarization, and reasoning.

3.3 Construction of Faulty Question Set

To assess the metacognitive abilities of the selected LLMs, we constructed a set of faulty questions across various science disciplines. Following the project guidelines, we initially generated a total of 3,000 faulty questions, comprising 500 questions in each of six disciplines: Mathematics, Physics, Biology, Chemistry, Earth Science, and Computer Science. Due to computational limitations and data processing constraints, we limited the experimental dataset to 600 questions, selecting 100 questions from each discipline. The selection process was based on cosine similarity analysis to ensure diverse coverage while minimizing redundancy.

3.3.1 Question Types and Categories

The faulty questions were designed to represent common error types that challenge LLMs' reasoning abilities. We categorized the questions into three main types:

1. Logical Errors: Questions containing logical inconsistencies or impossible scenarios.
2. Conceptual Errors: Questions involving misunderstandings or misapplications of fundamental concepts.
3. Calculation Errors: Questions that include impossible calculations or misuse of data.

3.3.2 Examples of Faulty Questions

Below are examples of faulty questions from the Mathematics discipline, along with explanations of why they are faulty:

1. Question: In a right triangle, one angle is 45° and another is 60°. Find the measure of the third angle.
   - Fault Reason: In a right triangle, one angle must be exactly 90°. The sum of the angles in any triangle is 180°. Adding the given angles ($90° + 45° + 60°$) results in 195°, which exceeds 180°, making the scenario impossible.
2. Question: A square swimming pool has a perimeter of 40 meters and an area of 120 square meters. Find the length of one side.
   - Fault Reason: A square with a perimeter of 40 meters has sides of 10 meters (since perimeter = 4 × side length). This would result in an area of 100 square meters (area = side length squared), not 120 square meters, creating a contradiction.

3. Question: John has 8 marbles and gives away half of them each day for 4 days. How many marbles does he have left?
   - Fault Reason: Halving the number of marbles each day leads to fractional marbles, which is not possible with discrete items like marbles. After four days, John would theoretically have 0.5 marbles, which is impossible.

3.4 Experimental Design and Evaluation Metrics

We designed three experiments to assess the metacognitive abilities of the selected LLMs: error detection, error admission and correction, and context maintenance. Each experiment focused on a specific aspect of the LLMs' reasoning capabilities, and appropriate evaluation metrics were developed to quantify their performance.

3.4.1 Experiment 1: Error Detection Ability

In the first experiment, we evaluated the models' ability to recognize errors in faulty questions and explicitly point them out. Each LLM was presented with the faulty questions individually, along with the instruction: "Provide your solution."

The models generated responses, which were recorded for analysis. We measured two key metrics:

1. Error Detection Rate (EDR): Defined as the proportion of faulty questions where the LLM detected and pointed out the error. The detection of errors was automated through pattern matching in the model's responses, where a response was considered to have detected an error if it contained any predefined error-indicating phrases such as "impossible", "contradiction", "invalid", "cannot exist", "inconsistent", etc. It is calculated as:

$$\text{EDR} = \left(\frac{\text{Number of questions where the error is detected}}{\text{Total number of faulty questions}} \right) \times 100\%$$

2. Error Identification Accuracy (EIA): Measures how accurately the LLM's identified error matches the actual fault reason. This accuracy was assessed using semantic similarity between the LLM's response and the actual fault reason. Specifically, we calculated EIA as the cosine similarity between the embeddings of the response and the actual fault reason:

$$\text{EIA} = \cos\left(\text{Embedding(Response)},\ \text{Embedding(Actual Reason)}\right)$$

   where Embedding($\cdot$) represents the sentence embedding function [17], and $\cos(\cdot, \cdot)$ denotes the cosine similarity function.

3.4.2 Experiment 2: Error Admission and Correction Ability

The second experiment assessed the models' ability to admit their own errors upon receiving feedback and to provide corrected answers. After the initial responses from Experiment 1, we provided each LLM with the following instruction: "Your previous answer to this problem may have contained errors. Please review the problem again, identify any potential mistakes, and provide a corrected answer if necessary."

If available, the previous response was included in the prompt to provide context. The models then generated revised answers, which were recorded. We evaluated:

1. Error Admission Rate (EAR): The proportion of instances where the LLM admitted its error upon receiving feedback, calculated as:

$$\text{EAR} = \left(\frac{\text{Number of times the LLM admits an error}}{\text{Number of errors pointed out}}\right) \times 100\%$$

2. Correction Accuracy (CA): The improvement in error identification accuracy after correction. It is calculated by comparing the EIA before and after the correction:

$$\text{CA} = \max\left(0,\ \frac{\text{EIA}_{\text{after}} - \text{EIA}_{\text{before}}}{\text{EIA}_{\text{before}}}\right)$$

where $\text{EIA}_{\text{before}}$ is the Error Identification Accuracy from the initial response, and $\text{EIA}_{\text{after}}$ is the Error Identification Accuracy after correction.

3.4.3 Experiment 3: Context Maintenance Ability

In the third experiment, we evaluated the models' ability to maintain consistency over repeated interactions and to recognize and correct initial mistakes without external feedback. The same faulty question was provided once more to each LLM, using the instruction: "Let's analyze this problem one more time. Please review it carefully and provide your analysis."

A conversation history was maintained to simulate a continuous dialogue. We measured the Context Consistency Score (CCS) to assess how similar the LLM's responses were over repeated interactions. The Context Consistency Score was calculated as the cosine similarity between the embeddings of the current response and the initial response:

$$\text{CCS} = \cos\left(\text{Embedding(Current Response)},\ \text{Embedding(First Response)}\right)$$

This metric reflects the degree to which the LLM maintains consistency in its responses over time.

3.4.4 Data Collection and Analysis Methods

We developed custom scripts to automate the data collection and analysis processes, ensuring consistency and efficiency across all experiments. In our testing environment, which consisted of an Intel Xeon Gold 5315Y (45GB CPU RAM) and an Ampere A4000 (16GB GPU RAM), the processing time for each LLM varied significantly. We utilized Python version 3.11.7 and the PyTorch version 2.1.1 deep learning framework for the experiments. Our transformer-based models and APIs were implemented using the Transformer 4.45.2 library. The dataset generation time for each LLM took anywhere from a minimum of 3 hours to a maximum of 17 hours. For each LLM, we conducted the following steps:

1. Initial Interaction: Presented each of the 600 faulty questions individually and recorded the LLM's responses for Experiment 1.

2. Feedback Provision (Experiment 2): Provided standardized feedback indicating potential errors in the LLM's initial answers and recorded subsequent responses.

3. Repeated Interaction (Experiment 3): Re-presented the same questions one time to observe any changes in responses, maintaining conversation history to simulate ongoing dialogue.

All interactions were meticulously logged, including prompts, responses, and any conversation history. Data were stored in structured formats to facilitate ease of analysis and ensure reproducibility. Each record included model name, question content, experiment type and responses.

We compiled the responses and computed the evaluation metrics outlined in Section 3.3. For error detection, we analyzed the responses for the presence of error-related terms and phrases. The EIA, CA, and CCS were calculated using the equations provided, utilizing sentence embeddings and cosine similarity measures. Also, statistical analyses were performed to determine the significance of differences between models, employing appropriate tests for categorical and continuous data.

4 Experimental Results

In this section, we present the results of our experiments designed to assess the metacognitive abilities of the selected LLMs. The experiments focused on three key areas: error detection ability, error admission and correction ability, and context maintenance ability. We evaluated each model using the metrics defined in Section 3, namely Error Detection Rate (EDR), Error Identification Accuracy (EIA), Error Admission Rate (EAR), Correction Accuracy (CA), and Context Consistency Score (CCS). Additionally, we analyzed the models' performance across different disciplines, including Mathematics, Physics, Biology, Chemistry, Earth Science, and Computer Science. Statistical significance was determined using non-parametric tests due to the non-normal distribution of the data.

4.1 Experiment 1: Error Detection Ability

4.1.1 Overview of Results

In the first experiment, we assessed each model's ability to recognize and explicitly point out errors in faulty questions. Table 4.1 summarizes the EDR and EIA for each model, and Table 4.2 presents the EDR for each model across different disciplines.

Table 4.1: Error Detection Ability Results

Model	EDR (%)	EIA (Average)
GPT-4o	16.5	0.5286
Claude 3.5 Haiku	8.33	0.5379
Llama 3.2-3B-Instruct	6.67	0.5245
Qwen 2.5-3B-Instruct	**21.67**	0.5149
Gemma-2-2B-It	7.00	0.5164

Table 4.2: Error Detection Rate by Discipline

Model	Mathematics	Physics	Biology	Chemistry	Earth Science	Computer Science
GPT-4o	**51.0**	13.0	12.0	10.0	7.0	6.0
Claude 3.5 Haiku	31.0	3.0	5.0	5.0	2.0	4.0
Llama 3.2-3B-Instruct	21.0	4.0	5.0	2.0	2.0	6.0
Qwen 2.5-3B-Instruct	37.0	33.0	16.0	16.0	14.0	14.0
Gemma-2-2B-It	22.0	2.0	4.0	7.0	2.0	5.0

4.1.2 Analysis and Discussion

Qwen 2.5-3B-Instruct achieved the highest overall EDR of 21.67%, indicating it was the most effective at recognizing faulty questions among the models. Notably, Qwen 2.5-3B-Instruct performed exceptionally well in Mathematics and Physics, with EDRs of 37.0% and 33.0%, respectively. GPT-4o also demonstrated strong performance, particularly in Mathematics with an EDR of 51.0%, which is the highest discipline-specific rate among all models. The EIA scores were relatively similar across all models, ranging from 0.5149 to 0.5379. This suggests that when the models detected errors, their explanations were of comparable quality in terms of semantic similarity to the actual fault reasons. To determine whether the differences in performance among the models were statistically significant, we employed the Kruskal-Wallis H-test. The results indicated statistically significant differences among the models' EDRs ($p < 0.001$).

Consider the following faulty question and the response from Claude 3.5 Haiku:

- Question: Find the area of a triangle with sides 5, 5, and 11.
- Claude 3.5 Haiku's Response: Using Heron's formula... However, this gives an imaginary number, which means these side lengths cannot form a triangle. To form a triangle, the sum of any two sides must be greater than the third side. $5 + 5 = 10$, which is not > 11. Therefore, a triangle with sides 5, 5, and 11 is impossible.

In this case, LLMs correctly identified that the triangle cannot exist due to the triangle inequality theorem, demonstrating successful error detection. This contributed positively to its EDR and EIA.

4.2 Experiment 2: Error Admission and Correction Ability

4.2.1 Overview of Results

The second experiment evaluated the models' ability to admit errors upon receiving feedback and provide corrected answers. Table 4.3 presents the EAR, CA, and the number of questions analyzed for each model, and Table 4.4 shows the EAR for each model across different disciplines.

Table 4.3: Error Admission and Correction Ability Results

Model	EAR (%)	CA (Average)	Questions Analyzed
GPT-4o	33.93	0.1303	501
Claude 3.5 Haiku	29.27	0.0987	550
Llama 3.2-3B-Instruct	13.75	0.0337	560
Qwen 2.5-3B-Instruct	30.21	**0.1470**	470
Gemma-2-2B-It	**40.04**	0.0449	557

Table 4.4: Error Admission Rate by Discipline

Model	Mathematics	Physics	Biology	Chemistry	Earth Science	Computer Science
GPT-4o	36.73	32.18	27.27	33.33	32.26	42.55
Claude 3.5 Haiku	50.72	30.93	20.00	35.79	23.47	20.83
Llama 3.2-3B-Instruct	13.92	11.46	9.47	16.33	19.39	11.70
Qwen 2.5-3B-Instruct	42.86	35.82	33.33	22.62	25.58	25.58
Gemma-2-2B-It	40.26	42.86	39.58	43.01	34.69	40.00

4.2.2 Analysis and Discussion

Gemma-2-2B-It demonstrated the highest overall EAR at 40.04%, indicating a strong tendency to admit errors upon feedback. GPT-4o and Qwen 2.5-3B-Instruct also showed high EARs of 33.93% and 30.21%, respectively. Llama 3.2-3B-Instruct had the lowest EAR at 13.75%, suggesting less propensity to acknowledge mistakes. In terms of Correction Accuracy, Qwen 2.5-3B-Instruct achieved the highest average CA of 0.1470, indicating better quality in the corrections provided after admitting errors. GPT-4o and Claude 3.5 Haiku followed with CA scores of 0.1303 and 0.0987, respectively.

An illustrative example from Claude 3.5 Haiku:

- Initial Question: The probability of event A is 0.4 and event B is 0.3. If $P(A \cap B) = 0.5$, find $P(A \cup B)$.
- Initial Response: Let's solve this step by step... $P(A \cup B) = 0.4 + 0.3 - 0.5 = 0.2$... The answer is 0.2.
- Feedback Provided: Your previous answer to this problem may have contained errors. Please review the problem again, identify any potential mistakes, and provide a corrected answer if necessary.
- Revised Response: Let me review the solution carefully... The given probabilities are inconsistent because $P(A \cap B)$ cannot be greater than $P(A)$ or $P(B)$... Therefore, there is no valid solution with these given probabilities.

In the revised response, LLMs admitted the inconsistency in the given probabilities and correctly concluded that the problem has no valid solution, demonstrating both error admission and correction.

4.3 Experiment 3: Context Maintenance Ability

4.3.1 Overview of Results

The third experiment assessed the models' ability to maintain consistency over repeated interactions without external feedback. Table 4.5 shows the CCS for each model, and Table 4.6 presents the CCS for each model across different disciplines.

Table 4.5: Context Maintenance Ability Results

Model	CCS (Average)
GPT-4o	0.9339
Claude 3.5 Haiku	0.9263
Llama 3.2-3B-Instruct	0.9399
Qwen 2.5-3B-Instruct	0.8977
Gemma-2-2B-It	0.9456

4.3.2 Analysis and Discussion

All models demonstrated high Context Consistency Scores, indicating strong ability to maintain consistent responses over repeated interactions. Gemma-2-2B-It achieved the highest overall CCS

Table 4.6: Context Consistency Score by Discipline

Model	Mathematics	Physics	Biology	Chemistry	Earth Science	Computer Science
GPT-4o	0.9401	0.9104	0.9424	0.9337	0.9470	0.9299
Claude 3.5 Haiku	0.9345	0.9320	0.9325	0.9064	0.9246	0.9280
Llama 3.2-3B-Instruct	0.9306	0.9185	0.9583	0.9272	0.9629	0.9417
Qwen 2.5-3B-Instruct	0.9148	0.9111	0.8924	0.8818	0.8938	0.8925
Gemma-2-2B-It	0.9324	0.9579	0.9541	0.9283	0.9508	0.9500

at 0.9456, closely followed by Llama 3.2-3B-Instruct at 0.9399 and GPT-4o at 0.9339.

From Claude 3.5 Haiku:

- (First Interaction)
  - Question: If $\sin(A) = 0.8$ and $\cos(A) = 0.8$, what is $\tan(A)$?
  - Response: Let's solve this step by step... $\tan(A) = \frac{\sin(A)}{\cos(A)} = \frac{0.8}{0.8} = 1$ The answer is 1.
- (Repeated Interaction)
  - Question: Let's analyze this problem one more time. Please review it carefully and provide your analysis.
  - Response: Upon re-examination, I notice that $\sin^2(A) + \cos^2(A) = (0.8)^2 + (0.8)^2 = 0.64 + 0.64 = 1.28$, which exceeds 1. This contradicts the Pythagorean identity $\sin^2(A) + \cos^2(A) = 1$. Therefore, the given values are inconsistent, and there is no real angle A that satisfies both conditions.

In the repeated interaction, LLMs sometimes recognized the inconsistency in the given values, correcting its initial mistake without external feedback. This demonstrates effective context maintenance and self-correction.

5 Discussion

In this section, we delve into an in-depth analysis of the experimental results presented in Section 4. We interpret the findings of each experiment, compare the performance of the models, and discuss the potential reasons behind their varying capabilities. We also address the limitations of our study and suggest directions for future research.

5.1 Interpretation of Results

The first experiment evaluated the models' ability to detect errors in faulty questions. Qwen 2.5-3B-Instruct achieved the highest overall EDR at 21.67%, indicating strong proficiency in recognizing faulty questions. Notably, Qwen 2.5-3B-Instruct excelled in Mathematics and Physics, suggesting that it may have been trained extensively in these domains. GPT-4o also demonstrated strong performance, particularly in Mathematics with an EDR of 51.0%, the highest discipline-specific rate among all models. This highlights GPT-4o's advanced capabilities in mathematical reasoning and error detection. The EIA scores were relatively similar across all models, indicating that when errors were detected, the quality of the explanations provided was comparable.

In the second experiment, we assessed how the models handle their own errors upon receiving feedback. Gemma-2-2B-It demonstrated the highest EAR at 40.04%, indicating a strong tendency to acknowledge mistakes. However, its CA was moderate compared to other models. Qwen 2.5-3B-Instruct achieved the highest CA at 0.1470, suggesting that not only does it admit errors, but it also provides high-quality corrections. GPT-4o and Claude 3.5 Haiku showed balanced performance in both EAR and CA.

The third experiment evaluated the models' consistency in responses when the same question was repeated without external feedback. All models exhibited high CCS, with Gemma-2-2B-It achieving the highest CCS at 0.9456. This demonstrates that even smaller models can effectively maintain context over multiple interactions.

5.2 Comparison of Model Performance and Analysis of Causes

GPT-4o exhibited strong error detection abilities, especially in Mathematics. Maintained high context consistency, indicating robust understanding and retention capabilities. Claude 3.5 Haiku demonstrated balanced performance across all experiments. Its ability to correct errors upon feedback reflects solid metacognitive skills. Llama 3.2-3B-Instruct showed exceptional context maintenance but lower error detection and admission rates, suggesting strengths in consistency but limitations in error recognition. Qwen 2.5-3B-Instruct achieved the highest EDR and CA, indicating proficiency in both detecting errors and providing quality corrections, particularly in technical disciplines like Mathematics and Physics. Gemma-2-2B-It had the highest EAR and CCS, showing a strong propensity to admit errors and maintain context, despite being a smaller model.

The differences in performance among the models can be attributed to variations in model size, architecture, and training data. Larger models like GPT-4o benefit from extensive training and complex architectures, enhancing their capabilities in reasoning and error detection. Gemma-2-2B-It's strong performance in EAR and CCS, despite its smaller size, suggests that efficient training techniques and optimization can enable smaller models to exhibit robust metacognitive abilities. This highlights the importance of quality over quantity in training data and the effectiveness of specialized training methodologies.

5.3 Limitations of the Study and Future Directions

Our study faced several limitations in its scope and methodology. The question set primarily focused on Mathematics and Sciences, potentially not capturing the models' full range of abilities across other domains. The evaluation metrics (EIA and CA) relied on semantic similarity measures, which may not fully reflect the nuances of model understanding. Additionally, the necessity for manual analysis such as using error-indicating phrases in detecting the errors of responses introduced potential biases despite automation attempts.

Future research should address these limitations through several key approaches. First, diversifying the question set across broader disciplines and complexity levels would enable more comprehensive assessment of metacognitive abilities. Second, developing enhanced training methods focused on self-monitoring and error correction capabilities would improve model performance. Third, implementing more dynamic dialogue systems could facilitate deeper reasoning processes. Finally, creating more objective evaluation metrics would better quantify model performance while reducing reliance on subjective interpretations.

6 Conclusion

This study examined the metacognitive abilities of five advanced LLMs: GPT-4o, Claude 3.5 Haiku, Llama 3.2-3B-Instruct, Qwen 2.5-3B-Instruct, and Gemma-2-2B-It. Focusing on error detection, error admission and correction, and context maintenance, we assessed their effectiveness in recognizing and addressing errors through experiments with faulty questions across multiple disciplines.

Findings show that while all models have metacognitive abilities, performance varied across tasks and domains. Advanced models like GPT-4o excelled in error detection and context maintenance, especially in mathematics, suggesting that complexity and training breadth enhance metacognition. Notably, smaller models like Gemma-2-2B-It also performed well in admitting errors and maintaining consistency, indicating that efficient training can enable compact models to develop effective metacognitive skills.

Discipline-specific variations emphasize the importance of domain-specific training data; models performed better in areas with more training exposure, highlighting the need for diverse datasets to improve generalizability. Despite positive results, limitations exist: challenges remain in ensuring consistent error detection and correction across varied contexts, and current evaluation metrics may not fully capture the models' nuanced understanding, indicating a need for refined assessment tools.

This research deepens our understanding of LLMs' metacognitive capacities, offering insights into their strengths and areas for improvement. Enhancing metacognition is crucial for developing AI systems that are accurate, reliable, and self-aware, thereby building user trust. Future research should diversify training data across disciplines, develop training approaches focusing on self-monitoring and error correction, refine evaluation metrics to better capture qualitative reasoning, and enhance interaction strategies with more dynamic dialogue systems to improve reasoning processes and user experience.

In conclusion, advancing LLMs' metacognitive abilities is vital for their effective deployment in applications where accuracy and reliability are paramount. By enabling AI to recognize and reflect upon its own reasoning, we move toward creating intelligent agents that support users more effectively and responsibly. This study lays a foundation for ongoing efforts to enhance the reliability and performance of large language models across various domains.

References

[1] OpenAI, "Gpt-4 technical report," 2023.

[2] "The claude 3 model family: Opus, sonnet, haiku. "[Online]. Available: `https://api.semanticscholar.org/CorpusID:268232499`.

[3] F. Lareyre and J. Raffort, "Ethical concerns regarding the use of large language models in healthcare," *EJVES Vascular Forum*, vol. 61, p. 1, 2024, ISSN: 2666-688X.

[4] J. H. Flavell, "Metacognition and cognitive monitoring: A new area of cognitive-developmental inquiry.," *American Psychologist*, vol. 34, pp. 906–911, 1979. [Online]. Available: `https://api.semanticscholar.org/CorpusID:8841485`.

[5] Y. Wang and Y. Zhao, "Metacognitive prompting improves understanding in large language models," *arXiv preprint arXiv:2308.05342*, 2023.

[6] A. Didolkar, A. Goyal, N. R. Ke, *et al.*, *Metacognitive capabilities of llms: An exploration in mathematical problem solving*, 2024. arXiv: `2405.12205 [cs.AI]`.

[7] T. Brown, B. Mann, N. Ryder, *et al.*, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, *et al.*, Eds., vol. 33, Curran Associates, Inc., 2020, pp. 1877–1901.

[8] Y. Nie, Y. Wang, and M. Bansal, "Analyzing compositionality-sensitivity of nli models," *ArXiv*, vol. abs/1811.07033, 2018.

[9] Y. Zhang, S. Sun, M. Galley, *et al.*, "Dialogpt : Large-scale generative pre-training for conversational response generation," in *Annual Meeting of the Association for Computational Linguistics*, 2019.

[10] E. Sulem, J. Hay, and D. Roth, "Do we know what we don't know? studying unanswerable questions beyond SQuAD 2.0," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4543–4548.

[11] H. Jiang and O. Nachum, "Identifying and correcting label bias in machine learning," in *International Conference on Artificial Intelligence and Statistics*, 2019. [Online]. Available: `https://api.semanticscholar.org/CorpusID:58006519`.

[12] M. T. Cox and A. Raja, *Metareasoning: Thinking about Thinking.* The MIT Press, 2011, ISBN: 0262014807.

[13] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv: Machine Learning*, 2017.

[14] Meta, *Llama 3.2*, Oct 2, 2024, 2024. [Online]. Available: `https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_2`.

[15] A. Yang, B. Yang, B. Hui, *et al.*, "Qwen2 technical report," *arXiv preprint arXiv:2407.10671*, 2024.

[16] G. Team, "Gemma," 2024. DOI: `10.34740/KAGGLE/M/3301`. [Online]. Available: `https://www.kaggle.com/m/3301`.

[17] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Conference on Empirical Methods in Natural Language Processing*, 2019.