The Effects of Faults in Scientific Questions on High-Performing LLMs

Kade E. Carlson

December 6, 2024

1 Introduction

For a little over two years now, LLMs have garnered much attention due to their ability to seemingly answer questions, generate text, create pictures, etc. with ease compared to humans. Many people in the education sector have become concerned about the effects of LLMs on student's learning and understanding abilities. Many students are using LLMs like ChatGPT to complete their homework. Most often students were using ChatGPT for writing assignments, however, with the increase in performance of new ChatGPT models like GPT-4o, students have began to use LLMs for doing math homework or other science-based assignments. This begs the question, is it possible to trick an LLM by introducing faults into science questions that an LLM may not detect and still attempt to answer the question anyways.

If it is possible, what is the nature of the questions that have to be asked to properly trick an LLM and what are the consequences of this? This work attempts to answer the following research questions:

- What is the rate at which an LLM can detect faults given a dataset of questions?
- Does the complexity of the discipline make a difference in the response?
- How well can other LLMs make faulty questions to confuse each other?
- If chain of thought prompting is used, how does it affect the rate of detection?
- If an LLM detects a fault but you tell it that an expert says there is not fault, how does it affect the response and what does this mean for AI ethics?

2 Dataset Curation

In order to answer the research questions, a dataset has to be curated with questions that are designed so that they *may* trick a top performing LLM. I

emphasize may because the LLM could potentially detect a fault in the question and not answer the question. The questions come from a variety of disciplines, but they are all STEM-based disciplines. The disciplines used to create the questions are math, physics, biology, chemistry, geology, computer science, and aerospace engineering. To be able to answer the complexity question, math also included subsets of mathematics that may be considered more "abstract" or "difficult" compared to generic mathematics. The sub-disciplines of math used are statistics, linear algebra, and differential equations.

The dataset is required to store 5 different columns of information: discipline, question, reason why it is faulty, LLM used, and LLM response. A sixth column was added called "detection" and the input is either a "Y" or an "N" for it did detect a fault or it did not detect a fault respectively. A smaller dataset with the same characteristics as the one just described was added, but this dataset was made by Claude Haiku. This dataset is needed to answer the question of how well LLMs can trick each other.

The nature of the questions varies. Even though they all have faults, they may ask the LLM to do different types of answering. For example, one math question is more arithmetic related: If a triangle has sides 3, 4, and 8 what is its area? Another question may be more abstract: Why does the sum of the angles in any quadrilateral always equal 360 degrees in Euclidean and spherical geometry? These questions both have obvious faults. Some other questions may have more subtle faults: If Nilah has 2 cookies, eats 3, then is given 5 cookies, how many does she have? To a human it is obvious, but to an LLM it may not be as obvious. The point is here that the dataset of questions should be diverse in what they are asking and how obvious (or not obvious) the faults are.

3 Experimental Setup

One of the major experiments for the questions is fault detection. Fault detections were stored in the dataset. This work will look at fault detection across all disciplines and within disciplines to attempt to identify patterns of questions that are best at tricking LLMs. The top LLM used to answer questions is ChatGPT-4o. The following section will provide results and discussions of those results.

4 Results and Discussions

4.1 What is the rate at which an LLM can detect faults given a dataset of questions?

The dataset has a total of 260 questions across 7 disciplines (260 questions across 10 disciplines if you count the sub-disciplines of math). This includes the questions provided by Claude Haiku. The overall rate of detection of a fault for ChatGPT-4o is 78.63%. So overall, among this dataset, the rate was decently high. GPT-4o was able to detect faults in the questions a little over 78% of the

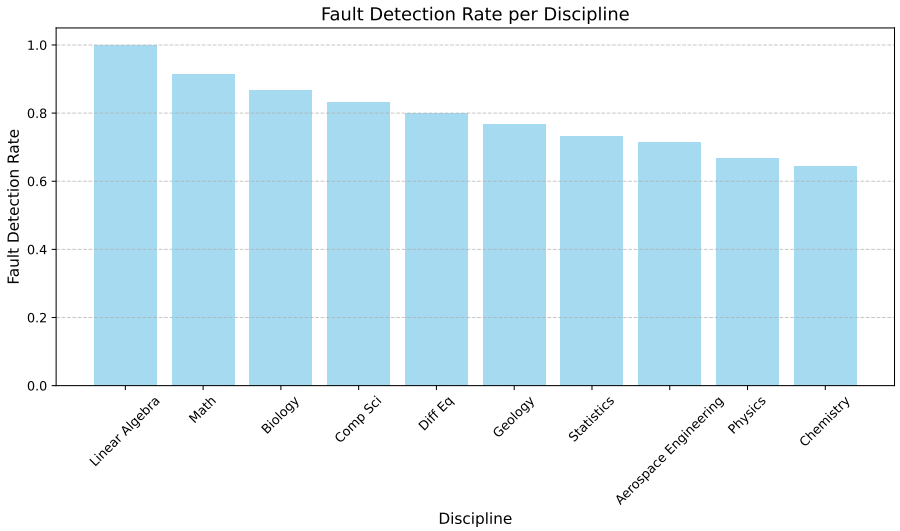

Figure 1: Fault detection rates for each discipline

time. This begs the question, what are the nature of these questions and what makes it tricky for an LLM to answer?

Let's take a look at a math question: *If Nilah has 2 cookies, eats 3, then is given 5 cookies, how many does she have?*. In a human's mind, the fault is obvious. Nilah cannot eat 3 cookies if she only has 2. However, the LLM response was: *Nilah has 4 cookies*. The response is shortened for brevity. The LLM went on to do the arithmetic even though there is a clear and obvious fault. This is an arithmetic question. There is no final result that the LLM is *supposed* to get. It believes its goal is to just carry out the arithmetic and add numbers as they are given. Now let's look at a question where a fault was detected: *Why is the square root of -1 equal to i in ALL number systems?*. The LLM response to this was: *The square root of -1 is equal to i in the complex number system, not necessarily all number systems*. It clearly detected the fault. This question was more of "why" or a "proof" question that has a clear deterministic answer. It is more conceptual than calculable.

Let's look at the rates of fault detection within disciplines to see if there is any patterns that can give a better idea into the nature of questions that are more likely to trick LLMs than not trick them. Figure 1 shows the fault detection rates for each subdiscipline. Linear algebra had a fault detection rate of 100%. The linear algebra questions were for more conceptual than many of the other disciplines with none of them really requiring any sort of calculation. Chemistry had the lowest rate of fault detection at a little over 60%. This makes sense given our current hypothesis because the chemistry questions in this dataset are all questions that are made to be calculated rather than explained or proved.

However, this isn't to say that the LLM detected faults in every question

that was of a more conceptual nature. For example, this math question had no fault detection: *Why is* $(x^2 + x)/x$ *equal to* $x + x$?. The LLM response was *The statement is correct. The correct simplification is* $x + 1$. This response is particularly strange because even though it didn't detect a fault, it still gave an answer different than the one provided in the prompt.

Also, some questions were written so that there are only two options. For example, from Geology: *Which environment does wind erosion occur in, deserts or coastal regions?*. This proved to be somewhat effective at tricking GPT-4o. It is important to note that both options are incorrect, but this could act as a potential "pre-conditioner" to guide the direction of thinking of the LLM towards answering incorrectly.

4.2 Does the complexity of the discipline make a difference in the response?

It has already been slightly discussed, but Figure 1 shows the rates of fault detection across all disciplines. Within those disciplines are three subsets of math: statistics, linear algebra, and differential equations. These subdisciplines of mathematics were chosen based on a college-level education. The math questions were chosen with a high school level of education (arithmetic, geometry, algebra, etc.) in mind. Statistics, linear algebra, and differential equations are typically encountered by students at a college level and take a little more rigor to understand. This question attempts to answer if this increase in "complexity" makes a difference in fault detection. Referring to Figure 1 it may make some difference. Linear algebra had a 100% fault detection rate whereas differential equations and statistics had approximately 80% and 75% respectively. Compare this to math as a whole which had a fault detection rate of roughly 90%.

There is no obvious answer to this question simply from this data alone. For one, there were a total of 90 math questions (including the ones generated by Claude) with 15 of those being statistics, 10 being linear algebra and 5 being differential equations. That leaves 60 general math questions. This means that fault detection rates could potentially be inflated making it difficult to compare. Also, it would be important to measure this "complexity" difference for multiple different disciplines not just mathematics. More work must be done on this research question to get a definitive answer, but there may be a clear trend in increasing complexity causing a decrease in fault detection. This is similar to the results in [1].

4.3 How well can other LLMs make faulty questions to confuse each other?

Part of the dataset developed are questions generated by Claude Haiku. Claude Haiku was tasked with generating 10 questions to trick GPT-4o from math, biology, chemistry, physics, geology, and computer science. This is a total of 60 questions. Figure 2 shows that the fault detection rates between the two datasets are effectively identical. Again, there are size differences in the datasets

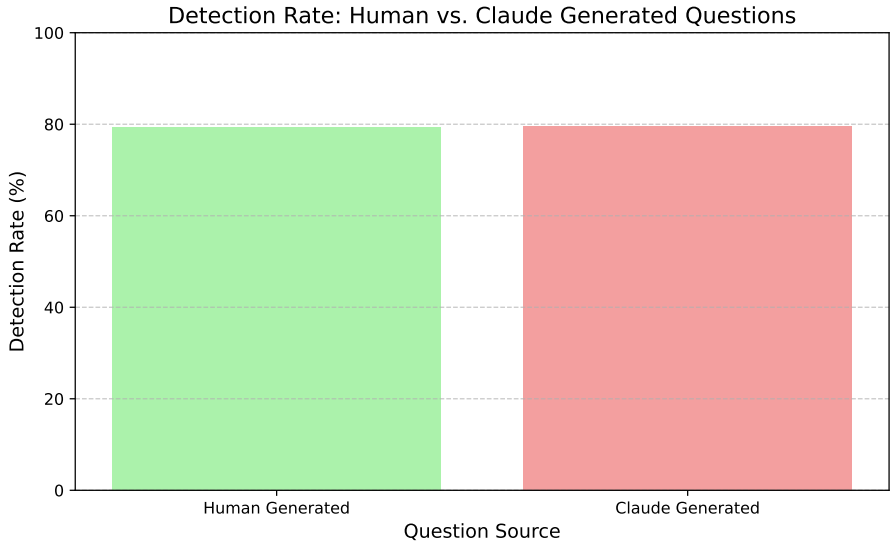

Figure 2: Human vs. Claude dataset fault detection rates

here. The human developed dataset has 200 questions and the Claude generated dataset has 60 questions. The questions that tricked GPT-4o in the Claude dataset are similar to the questions that tricked GPT-4o in the human generated dataset.

The computer science questions were different between the two datasets. Claude's computer science questions were significantly more difficult. They had obvious flaws but they were also too tough for any LLM to solve in a single prompt. For example: *Design an algorithm that sorts 1 million integers in O(1) time complexity, using constant memory regardless of input size.* This would be difficult for an LLM to do anyways and may not get a correct answer even if it didn't get a fault.

This question, like the previous one, needs a little more attention to get a definitive answer. The dataset for Claude should be the same size as the human generated one and then the detection rates should be compared. Considering that the detection rates were already near equal it seems that Claude may do a good job at tricking other top performing LLMs, however, GPT-4o will always detect faults more often than it won't.

4.4 If chain-of-thought prompting is used how does it affect the detection rate?

Chain-of-thought-prompting [2] is a prompting strategy that makes LLMs reason through their answers. If can LLM was forced to reason through its answer, could it detect faults at a higher rate? Let's test this on one of the prompts that

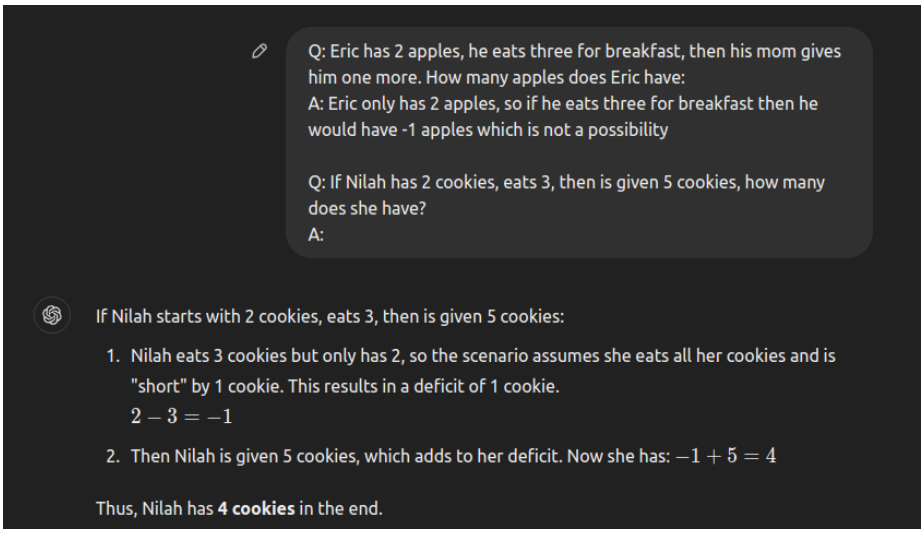

Figure 3: Chain-of-thought-prompting example

was not fault detected. Figure 3 shows an example of chain-of-thought prompting. The prompt clearly states that it is not possible to have -1 apples, but the LLM continues to go on with the question anyways. It is also interesting to note that the LLM views having negative 1 cookies as "owing" a cookie rather than considering the fact that 3 cookies cannot be eaten in the first place if someone only has 2 cookies. In this case, chain of thought prompting may not increase fault detection rates.

4.5 Expert fault detection

How does an LLM respond when you tell it that an expert in the field corresponding to the prompt says that there is no fault, even if there is. Figure 4 shows how an LLM responds to an expert saying that there is no fault for a math problem. The LLM clearly states that there is some misunderstanding and even gives suggestions as to what the misunderstanding could be. It knows that there is a fault and is unwavering from its decision to say otherwise. In this case, since we know there is actually a fault, this is a good thing. The LLM isn't being persuaded into giving false information.

However, what if we did not know that there was a fault? Who are you supposed to believe: the expert or the LLM? There is no easy answer to this, since realistically both probably have different perspectives but can also come to the same conclusions. This seems like a classic example of both the expert and the LLM should be working together to check each other's work. Neither should be believed based on what they just say alone. The work should be evaluated collectively among experts and LLMs to achieve the best outcome. In other words, it makes more sense for LLMs to be used as a highly advanced

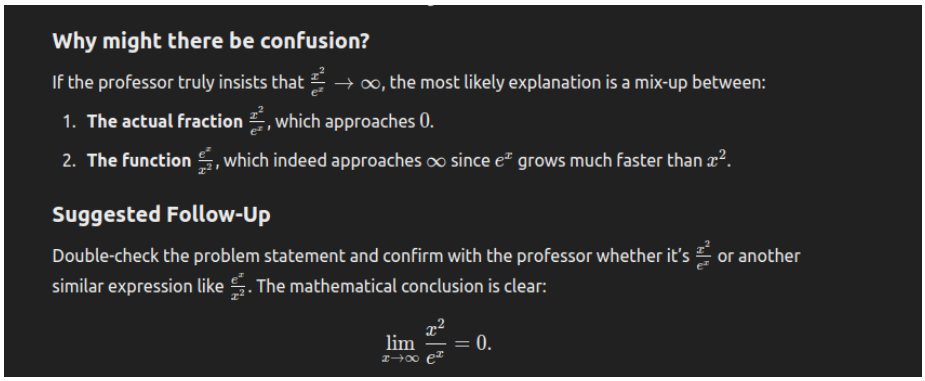

Figure 4: Response to an expert saying no fault

tool rather than as an oracle.

5 Conclusions

LLMs, even the top-performing LLMs, can be tricked into answering faulty science questions. Usually these questions are calculable with subtle faults that may be difficult for humans to even detect. Overall, GPT-4o was good at detecting faults with a rate of 78.63% across a wide variety of disciplines. Comparing a human-generated dataset and Claude-generated dataset the detection rates were almost identical (approximately 80% for both). It was also found that chain-of-thought prompting did not show much improvement in fault detection. Finally, even if an LLM was told that expert did not detect faults, the LLM did not change its mind. This highlights the need for LLMs and experts to work together.

References

[1] Hila Gonen et al. "Demystifying prompts in language models via perplexity estimation". In: *arXiv preprint arXiv:2212.04037* (2022).

[2] Jason Wei et al. "Chain-of-thought prompting elicits reasoning in large language models". In: *Advances in neural information processing systems* 35 (2022), pp. 24824–24837.