

# Are LLMs Smarter than an Eighth Grader? Exploring Mathematical Abilities of Large Language Models

John-Wesley Appleton and Tin Do and Edmund Doerksen and William Qi

## Abstract

This project investigates the mathematical reasoning capabilities of large language models (LLMs) by evaluating their performance on the GSM8K dataset, a collection of grade 8-level math word problems. While LLMs have demonstrated high accuracy on this dataset, we hypothesize that their success stems from memorization rather than genuine reasoning abilities. To test this, we introduced two modified datasets: one with altered numerical values and entities, and another with irrelevant contextual information added. Performance on these datasets significantly declined compared to the original, indicating limited reasoning skills. Error analysis revealed that stronger models often made calculation errors despite following correct reasoning steps, while weaker models were misled by irrelevant details. Our findings suggest that LLMs' reported success on GSM8K is more reflective of memorization than true mathematical proficiency.

## 1 Introduction

At present, state-of-the-art LLMs demonstrate astounding accuracy across a variety of natural language processing tasks, matching or even surpassing human performance. However, these models are generally trained and tested on carefully prepared data, which do not match the diverse nature of inputs in the real world. Right now, language models do not seem to be able to perform genuine logical reasoning, especially on math problems. To improve models' ability to handle diverse inputs, a natural next step is to attain the capability to reason.

We reimplement a recent publication by Apple Research, where a team of scientists investigated the performance of SOTA LLMs on elementary- and middle-school level math word problems. The researchers generated variations of the problems that did not change the strategy required for solving them: they switched out the numbers involved and added extra clauses. They found that

these changes, although semantically meaningless, greatly reduced performance, with many models experiencing an over 30% drop in accuracy.

The problem that we are trying to solve is to evaluate and compare the performance of several state-of-the-art models on a modified GSM8K dataset. Given an example from the GSM8K test set, we will create parseable templates, as shown in the figure below. These templates are used to create modifications to the original GSM8K question and evaluate the performance of the LLMs on these variations.

```
When {name} watches her {family}, she gets out a variety of toys for him. The bag of building blocks has {x} blocks in it. The bin of stuffed animals has {y} stuffed animals inside. The tower of stacking rings has {z} multicolored rings on it. {name} recently bought a tube of bouncy balls, bringing her total number of toys she bought for her {family} up to {total}. How many bouncy balls came in the tube?
```

```
#variables:
- name = sample(names)
- family = sample(["nephew", "cousin", "brother"])
- x = range(5, 100)
- y = range(5, 100)
- z = range(5, 100)
- total = range(100, 500)
- ans = range(85, 200)

#conditions:
- x + y + z + ans == total
```

---

```
Let T be the number of bouncy balls in the tube. After buying the tube of balls, {name} has {x} + {y} + {z} + T = {x + y + z} + T = {total} toys for her {family}.
```

```
Thus, T = {total} - {x + y + z} = <<{total}-{x + y + z}>>{ans}>>{ans} bouncy balls came in the tube.
```

In this project we will replicate the increasing levels of difficulties that the research paper mentioned. More specifically, we will attempt adversarial input generation strategies like changing the specific numbers and adding irrelevant information. These additions are designed not to affect the reasoning needed to solve the problem but might distract models.

Formally, the problem is to see how much LLM performance is degraded upon varying the exact

wording of math questions. By evaluating the accuracy difference, we will be able to quantify how much different models are affected by these perturbations, which will give insight into how much these models overfit to the GSM8K dataset as well as their ability to reason through “new” problems.

We chose this task as our term project because we believe that reasoning in LLMs is one of the next big breakthroughs that will significantly improve their performance. LLMs have been known to struggle with math problems due to lack of reasoning ability, and we wanted to see if the high accuracy on GSM8K of many models is simply an artifact of them being trained on the dataset instead of actually being able to solve the problems.

## 2 Literature Review

Several recent works have investigated the ability of LLMs to adapt to variants of GSM8K problems, demonstrating performance drops each time.

**Large Language Models Can Be Easily Distracted by Irrelevant Context.** In this study, the authors worked with a dataset—GSM-IC—constructed by taking 100 problems from GSM8K and then adding exactly one irrelevant sentence to each of them. GSM-IC was then blown up in size to nearly 60,000 values by creating dozens of variants of the original 100 problems (each with a slightly altered irrelevant sentence). Variation in the irrelevant sentences was achieved through changing the names that appeared in them and whether or not these names referenced people mentioned in the original problem. The models used were GPT 3.5 and Codex, and the authors demonstrated that these LLMs experienced performance deterioration when attempting to solve the new math problems presented to them. Even with state of the art prompting techniques like chain-of-thought and prompting with programs, accuracy on variants of problems that the models had been able to solve correctly unmodified plummeted to 18%.

**GSM-PLUS: A Comprehensive Benchmark for Evaluating the Robustness of LLMs as Mathematical Problem Solvers.** In this work, the authors generated GSM8K variants to confuse two dozen state of the art LLMs. They employed GPT-4 to construct the variants, which altered the original problems by changing numbers, adding irrelevant context, and rephrasing text without changing meaning. GPT-4 proved unreliable in generating problems that were of appropriate similarity to

those in GSM8K, so human validation was necessary to complete the preparation of the new dataset, called GSM-PLUS. The authors employed compositional prompting, which breaks down problems into iterative steps when the LLMs are queried, but still saw dramatic decreases in performance on GSM-PLUS vs GSM8K. Accuracy drops occurred even for models such as Abel and ToRA that are fine-tuned on mathematical datasets. The models were particularly susceptible to errors when exposed to problems containing arithmetic variation and distractor insertion, indicating a reliance on pattern recognition over genuine understanding.

**GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models.** This study’s authors assessed the ability of 25 LLMs to perform mathematical reasoning. Given the empirical evidence that LLMs are highly sensitive to the input tokens they receive, the authors believed that perturbations in GSM8K would result in significant accuracy drops. So, they created the 5000-problem GSM-Symbolic dataset. This new dataset featured 50 batches of 100 questions each. Each batch was generated from a single GSM8K problem by altering keywords and changing numbers in different ways while leaving unchanged the sequence of mathematical steps needed to arrive at a correct solution. The authors’ approaches demonstrated notable drops in accuracy, with the greatest changes occurring when questions had both their keywords and numbers altered. An additional experiment was conducted through a separate dataset created by adding irrelevant clauses to GSM8K variants, which resulted in a whopping 45% accuracy drop across the LLMs on average.

## 3 Experimental Design

To evaluate the performance of large language models with approximately 3 billion parameters, we conducted experiments comparing their accuracy on the original GSM8K dataset and two modified variants of the dataset. The original GSM8K dataset, which contains grade-school math problems, served as the baseline for evaluating model performance. The six models we evaluated were OpenAI’s GPT-40, GPT-40 mini, Meta’s Llama-3.2-3B-Instruct, Google’s Gemma-2-2b-it, Microsoft’s Phi-3.5-mini-instruct, and Qwen’s Qwen2.5-3B-Instruct.

### 3.1 Dataset Modifications

Two new datasets were created to introduce controlled variations and measure the robustness of the models under different conditions:

- **Numerical Alterations:** The first dataset was constructed by modifying 54 original GSM8K questions to create 2,700 variants. These modifications involved altering the numerical values and entities in the questions to test the models' ability to handle numerical variations.
- **Irrelevant Additions:** The second dataset consisted of the same 2,700 questions as the first but with an additional irrelevant sentence appended to each question. This dataset was designed to assess the impact of irrelevant context on model performance, especially when combined with numerical alterations.

### 3.2 Evaluation Procedure

Each model was evaluated on three datasets:

- A randomly sampled set of 2,700 questions from the original GSM8K dataset.
- The modified dataset with numerical alterations.
- The modified dataset with both numerical alterations and irrelevant context additions.

The evaluation metric was **accuracy**, defined as the fraction of responses generated by the models that matched the correct answers. To ensure fair evaluation, we implemented a post-processing step to extract the final numerical value from each model's response. The process involved parsing the response text using a regular expression to identify numerical values, selecting the last numerical value found, and converting it to either a float or an integer depending on its format. The numbers of commas were removed to handle cases such as thousands of separators and the extracted value was compared to the ground truth. A response was considered correct if the extracted number was within a tolerance of 0.05 of the ground truth value. This approach accounted for variations in numerical precision, such as differences in decimal places, ensuring that slight discrepancies did not unfairly penalize the models.

By comparing model accuracy across these datasets, we aimed to quantify the models' resilience to numerical changes and irrelevant context, providing insights into their robustness and

reliability in handling modified and noisy input data.

## 4 Results

### 4.1 Published Baseline

For our project, we used GPT-4o as our strong baseline. This is OpenAI's flagship model and believed to be one of the best models right now. Apple's GSM-Symbolic paper obtained an accuracy of 95% on GSM8K with GPT-4o. Using the evaluation metric mentioned earlier, we also obtained a similar accuracy, obtaining a 95.56% accuracy on 2700 randomly sampled questions from GSM8K. Our results are directly comparable as we use the same test set (GSM8K), and it is encouraging to see that we were able to obtain very similar results to them, as it meant that our code implementation was correct and consistent with their research. This also means that we can have a direct comparison with the other models as well.

### 4.2 Extensions

The models' performance across the three datasets (unaltered + two extensions) is shown in [Table 1](#). GPT-4o easily outperforms the other models on all three datasets. Its most impressive result is a 0.8974 accuracy on the GSM8K problems with numbers and entities altered, where it is 25 points ahead of the next best model. However, it does experience a significant drop once irrelevant context is introduced, which is unlike all of the other models.

Overall, we see that the first extension (changing numbers and entities) consistently decreased model performance. For all models other than GPT-4o, the drop is quite significant, with most of them experiencing a 20-30% decrease in accuracy. This is surprising, but it reveals that most LLMs likely do not have great mathematical reasoning abilities and are simply overfitting to the GSM8K benchmark.

For extension two (changing numbers/entities and adding irrelevant context), the models further experienced a drop in accuracy. For most models, though, it was only a small decrease in accuracy compared to extension 1. This means that adding irrelevant context doesn't affect model performance as much as changing the numbers/entities does, signifying that they are quite good at parsing through irrelevant information.

Model	GSM8K	GSM8K + Numbers/Entities	GSM8K + Numbers/Entities/Irrelevant Context
GPT-4o	0.9556	0.8974	0.7019
GPT-4o mini	0.8570	0.6463	0.6330
Llama-3.2-3B-Instruct	0.8426	0.4900	0.4922
Gemma-2-2b-it	0.6926	0.3867	0.3585
Phi-3.5-mini-instruct	0.8733	0.6078	0.5841
Qwen2.5-3B-Instruct	0.8800	0.5833	0.5693

Table 1: Performance of various models on GSM8K and its variants

### 4.3 Error Analysis

A particularly fascinating error class is displayed by our strong baseline, GPT-4o, where the convoluted nature of some word problems causes it to hallucinate. In example 3 of Table 2 in the Appendix, the model becomes distracted by one aspect of the problem (an improbable feeding regimen for the chickens), launching into a 500-word odyssey of a response that begins by attempting to address the problem but soon veers into nonsensical territory, generating sentences by combining words related to statistics and scientific research at random. This error does not appear to be common, however. In fact, for the 99 other variants of the chickens problems we tested, there was an error rate of less than 10%, with a long-winded hallucination such as this one happening just twice. Given the limited prevalence of the run-on hallucination, ensuring consistently correct responses from the LLMs (at least on the chicken problem) would be a simple matter of introducing self-consistency in prompting, whereby multiple chains of reasoning could be attempted, with only the most prevalent result selected as the model’s final response.

Another common error type as shown in Table 2 of the Appendix is when models are able to get all the steps correct but get a mathematical calculation incorrect. This is especially prevalent in stronger models like Phi-3.5-mini-instruct. This error is interesting because the model is shown to be strong enough to reason about the steps but being a language model and not a calculator, cannot accurately perform calculations especially when long decimals are involved.

Lastly, for weaker models, we saw that irrelevant context did indeed impact its reasoning abilities. Unable to recognize which information is useful and which information is noise, the model will incorporate the information of the irrelevant sentence into the final answer. Gemma-2-2b-it, which was our weakest model, commonly exhibited this error class. In the example in Table 2, we see that even though June’s sales are included in the question, the question only asks about April and May sales, but the model uses June’s sales as well in its final calculation.

## 5 Conclusions

Our analysis highlights the limitations of large language models in performing grade 8-level mathematical reasoning tasks. While these models achieve high accuracy on benchmark datasets such as GSM8K, their performance declines significantly when faced with modified datasets involving numerical alterations or irrelevant contextual information. This suggests that their success is primarily driven by memorization rather than genuine problem-solving skills. Error patterns further reveal that even stronger models struggle with basic mathematical operations, while weaker models are easily distracted by extraneous details. These findings emphasize the need for improved model architectures and training approaches to enhance reasoning capabilities, particularly in mathematical domains. Future work should focus on developing models that can generalize across variations in problem structures, ensuring robust and reliable mathematical understanding.

## 6 Acknowledgements

We would like to thank Professor Mark Yatskar for delivering engaging lectures that provided the foundational knowledge for this project. We are especially grateful to our teaching assistant, Ugurcan Vurgun, for his consistent support and guidance throughout the project. His regular meetings and constructive feedback were invaluable in helping us navigate challenges and refine our work. Their contributions have been instrumental to the completion of this project.

## 7 References

Li, Q., Cui, L., Zhao, X., Kong, L., Bi, W. GSM-PLUS: A Comprehensive Benchmark for Evaluating the Robustness of LLMs as Mathematical Problem Solvers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2961-2984, August 2024. Association for Computational Linguistics.

Mirzadeh, I., Alizadeh, K., Shahrokhi, H., Tuzel, O., Bengio, S., Farajtabar, M. GSM-Symbolic: Understanding the Limitations of Mathematical Reasoning in Large Language Models. *arXiv preprint arXiv:2410.05229v1*, 2024.

Shi, F., Chen, X., Misra, K., Scales, N., Dohan, D., Chi, E., Schärli, N., Zhou, D. Large Language Models Can Be Easily Distracted by Irrelevant Context. *arXiv preprint arXiv:2302.00093v3*, 2023.

## A Appendix

	Incorrect Response	Correct Solution
<b>Error 1: Model Hallucination (Both Extensions)</b>	It seems there is some confusion in the problem statement. Initially, it is mentioned that Olivia feeds each of her chickens 24 cups of mixed chicken feed per day, which would be an unusually large amount per chicken. However, based on the information provided, let's focus on calculating how much feed is needed for the final meal if the flock as a whole receives a certain total amount each day. From the next pieces of information, we know:- Olivia feeds her chickens a total of $32 + 43$ cups = 75 cups across the morning and afternoon meals. If we assume that Olivia has to feed a total of 24 cups per chicken, this contradicts the numbers, as no total number of chickens is provided. . . . Thus, based each method: encourage reviewer confirmed clarification given circumstances described, recommending kindly verifying direct local explorer sourced number verification spanning results before new cases absent detailed limitations apply for this deduction missing content.	If each chicken eats 24 cups of feed per day, then for 45 chickens they would need $24 \times 45 = \langle 24 \times 45 = 1080 \rangle 1080$ cups of feed per day. If she feeds the flock 32 cups of feed in the morning, and 43 cups in the afternoon, then the final meal would require $1080 - 32 - 43 = \langle 1080 - 32 - 43 = 1005 \rangle 1005$ cups of chicken feed. ##### 1005
<b>Error 2: Correct Reasoning but Incorrect Calculations (Both Extensions)</b>	To calculate Amelia's earnings, we need to convert the minutes she worked into hours since her pay rate is per hour. There are 60 minutes in an hour, so 29 minutes is $29/60$ hours. Now, we multiply the hours she worked by her hourly rate: $(29/60) \text{ hours} \times \$25/\text{hour} = \$11.875$ Since it's not common to pay in fractions of a cent, we can round this to the nearest cent. Amelia earned approximately \$11.88 for the babysitting she did yesterday. The answer is \$11.88.	Amelia earns $25/60 = \langle 25/60 = 0.4167 \rangle 0.4167$ per minute. Working 29 minutes, she earned $0.4167 \times 29 = \langle 0.4167 \times 29 = 12.0833 \rangle 12.0833$ . ##### 12.0833
<b>Error 3: Distracted by Irrelevant Context (Extension 3)</b>	Here's how to solve the problem: * **May Sales:** Amelia sold half as many clips in May as she did in April, so she sold $180 / 2 = 90$ clips. * **June Sales:** She sold three times as many clips in June as in April, so she sold $180 \times 3 = 540$ clips. * **Total Sales:** In total, Amelia sold $180 + 90 + 540 = 810$ clips. * **Answer:** Amelia sold a total of 810 clips in April and May.	Amelia sold $180/2 = \langle 180/2 = 90 \rangle 90$ clips in May. Amelia sold $180 + 90 = \langle 180 + 90 = 270 \rangle 270$ clips altogether in April and May. ##### 270

Table 2: LLM Error Examples on Changed Number and Irrelevant Context Datasets