



# GSM-Identity: Evaluating Mathematical Reasoning in LLMs via Equivalence Transformations

Kajal Negi<sup>1,2</sup> · Giovanni Puccetti<sup>1</sup> · Andrea Esuli<sup>1</sup>

Received: 3 April 2025 / Revised: 10 February 2026 / Accepted: 10 March 2026  
© The Author(s) 2026

## Abstract

We introduce GSM-Identity, a pipeline to modify existing mathematical reasoning benchmarks by adding extra complexity to the questions while preserving their fundamental meaning. By systematically transforming numerical values in the GSM8K dataset into mathematically equivalent but less obvious expressions, we create a benchmark to measure Large Language Models (LLMs) mathematical understanding. We evaluate LLMs ranging from 7 billions to 72 billions parameters using multiple prompting strategies, including standard, notice-based, and chain-of-thought approaches. We find that Math oriented models can retain most of their performance on GSM8K when evaluated on GSM-Identity, while general purpose models show significant performance degradation. A comparison with human evaluations reveals that models in the 7 billion parameters range perform similar to humans when exposed to the kind of modifications we study, while models with more than 70 billion parameters are more accurate than humans in answering the questions and they are also more resilient to modifications. Our findings highlight GSM-Identity as a valuable tool for distinguishing reasoning from memorization, offering insights into the abilities of LLMs to understand higher level mathematical concepts.

**Keywords** Large language models · Mathematical understanding · Reasoning · AI and Human

---

Editors: Riccardo Guidotti, Anna Monreale, Dino Pedreschi.

---

✉ Kajal Negi  
kajal.negi@isti.cnr.it

Giovanni Puccetti  
giovanni.puccetti@isti.cnr.it

Andrea Esuli  
andrea.esuli@isti.cnr.it

<sup>1</sup> Institute of Science and Technologies of Information “A. Faedo”, CNR, Pisa, Italy

<sup>2</sup> University of Pisa, Pisa, Italy

## 1 Introduction

Research on Generative AI has seen a fast growth in recent years due to performance improvements in generating each of the main modalities: text, images, videos and audio. This growth has been favored by the ability of Generative AI to create seemingly new content, i.e., ChatGPT-like models write coherent and grammatical text potentially leading to genuinely novel information.

Despite this potential, most use cases for Generative AI involve facilitating the interaction between users and existing technologies. For audio generative models both turning voice into text through Automatic-Speech-Recognition (ASR) models (Baevski et al., 2020) and the other way around through Text-To-Speech (TTS) models (Li et al., 2023) show promising ground for new applications. Similarly, vision generative models have found several applications in manipulating existing images and videos, for example through in-painting (Lugmayr et al., 2022) that have been added to established image modification tools.<sup>1</sup> Text generative models have also been used as assistants, most notably in code writing (Hui et al., 2024).<sup>2</sup>

One of the main challenges in creating truly novel content is the uncertainty of Generative models outputs, and the difficulty to quantify the performance of generative models in tackling a specific task. Indeed, while accuracy on existing benchmarks can approximate a model performance on a specific task, it is hard to understand if this is the result of generalization or memorization (Hupkes et al., 2023).

In an attempt to understand the ability of Large Language Models (LLMs) to generalize to unseen data, several benchmarks have been developed over time and they have been collected in large suites used to study LLMs abilities in performing a large share of the intellectual activities carried out by humans (Srivastava et al., 2023).

Recently, there have also been claims of strong reasoning abilities shown by LLMs, specifically from so called “Reasoning Models” such as DeepSeek-R1 and ChatGPT-o1. This is generally tested on complex existing math benchmarks such as GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). There is however concern that part of the performance might be related to dataset leakage between benchmark datasets and web-scale training sets of LLMs (Xu et al., 2024). E.g. Grattafiori et al. (2024) train on 15 trillions tokens, which cannot be effectively tested for data leakage.

Research on avoiding spurious results due to data leakage has developed mainly into two approaches: (a) adding modifications to existing benchmarks to measure how well models can correctly answer questions that appear differently from their original version, possibly indicating data leakage and limited reasoning/understanding skills (Mirzadeh et al., 2025), (b) creating synthetic benchmarks that are developed at test time and thus are in principle not learnable through scraping, to obtain an uncontaminated measure of LLMs’ reasoning ability (Zhou et al., 2025).

In this work we follow an approach that involves the modification of existing benchmarks. Our approach is novel in that it focuses on making existing questions seemingly more challenging, while actually leaving them unchanged at an abstract level. We modify the widely used GSM8K dataset (Cobbe et al., 2021) introducing complex mathematical

---

<sup>1</sup>Adobe Firefly.

<sup>2</sup>GitHub Copilot.

formulas to replace data in the questions, without modifying the value when evaluated, thus leaving the questions' meaning unchanged. The main contributions of this work are three:

1. GSM-Identity: A method to create new more challenging math benchmarks while keeping conceptual difficulty low to test LLMs reasoning abilities.
2. A comparative analysis of several LLMs, including state of the art models in math and coding, on our newly developed benchmarks.
3. A comparative analysis between humans and models on their ability to handle seemingly harder but conceptually equal questions along with an analysis of the performance drop.

## 2 Related Works

LLMs have shown the ability to perform most NLP tasks in a generative way, i.e., prompting them with requests and instructions about the task of interest, e.g. asking an LLM to “Summarize the following text:...” can be sufficient to perform text summarization, classically tackled with specialized systems. Since several NLP tasks are prone to be formulated through natural language, in principle, this makes LLMs able to tackle all the tasks at once (Brown et al., 2020; Radford et al., 2019).

Moreover, when explaining a task is not sufficient for an LLM to complete it, providing examples can improve performance significantly (Dong et al., 2024).

This level of generality leads to attribute *reasoning* abilities to LLMs, although the concept of reasoning has several interpretations depending on the context where it is cited (Choudhury et al., 2022; Mitchell, 2021). However, what mechanism allows LLMs to tackle a large number of tasks is still an open question.

Therefore, to assess the reasoning abilities of LLMs, different approaches have been studied. A prominent solution is the development of math-focused question answering benchmarks. Some of the most used benchmarks are GSM8K for grade school math (Cobbe et al., 2021), along with cleaner versions of the same dataset GSM8K-platinum (Vendrow et al., 2025), MATH for high-school math (Hendrycks et al., 2021) and Putnam-AXIOM for competition level math (Gulati et al., 2024).

While testing on math related benchmarks is an effective way to measure LLMs reasoning abilities, we cannot be sure that this is a fair measure of LLMs performance, due to potential data leakage. If data leakage is the reason behind LLMs performance, then their ability to perform several NLP tasks would not imply reasoning, but rather memorization (Bender et al., 2021).

While conceptually simple, it is hard to investigate data leakage in LLMs through directly checking for the presence of samples in the LLM training sets. Indeed, the large training corpora used to build modern language models, spanning trillions of tokens, are virtually impossible to inspect for contamination issues. As a solution, a new line of work is growing in recent years focused on modifying the questions in several ways to increase the difficulty and test reasoning abilities more accurately.

Specifically, there has been a focus on modifying questions in the GSM8K dataset, and other math-related benchmark datasets, to make it more challenging and test different reasoning patterns in LLMs, among others:

- GSM-IC (Shi et al., 2023) adds irrelevant context to GSM8K questions to test if the models can isolate meaningful information and use to answer the questions, e.g.; “Mary has 3 times as much candy as Megan. Mary then adds 10 more pieces of candy to her collection. If Megan has 5 pieces of candy, how many does Mary have in total?” becomes “Mary has 3 times as much candy as Megan. Mary then adds 10 more pieces of candy to her collection. Mary’s mother has 8 more books than Mary. If Megan has 5 pieces of candy, how many does Mary have in total?”
- GSM-Plus (Li et al., 2024) applies several kinds of modifications, including minor arithmetic changes to create a new version of GSM8K which is more general from several perspectives, the closest to our work is “Janet’s ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder at the farmers’ market daily for 3 per fresh duck egg. How much in dollars does she make every day at the farmers’ market?”;
- GSM-symbolic (Mirzadeh et al., 2025) which extract patterns from GSM8K questions to create equivalent versions where several details in each, to test strictly for memorization “Benny saw a 10-foot shark with 2 6-inch remoras attached to it. What percentage of the shark’s body length is the combined length of the remoras?” becomes “Sanjay saw a 60-foot dolphin with 16 12-inch remoras attached to it. But a quarter of the remoras go away. What percentage of the dolphin’s body length is the combined length of the remaining remoras?”;
- GSM-infinite (Zhou et al., 2025) generate very long, complex graph-based questions similar to the ones of GSM8K at test time, in order to avoid data leakage.<sup>3</sup>

More recent works (Huang et al., 2025) attempt to increase the difficulties of the MATH benchmark, inherently more challenging than GSM8K, by perturbing the questions to make the answers less obvious either without modifying the underlying reasoning (MATH-P-Simple) or by modifying it (Math-P-Hard). The MATH<sup>2</sup> benchmark (Shah et al., 2024) is also developed to make the MATH benchmark more difficult by leveraging LLMs to extract from the existing dataset the different skills needed to answer the questions, then mixing the extracted skills during the generation process to create new more complex questions.

Along with newer benchmarks, the development of stronger LLMs with reasoning skills has surged [DeepSeek-R1, Deepseek (2025)], also focusing on LLMs dedicated to math reasoning [Qwen Math, Yang et al. (2024)].

Our approach leverages ideas from both GSM-Plus and MATH-perturb, as we aim to diversify GSM8K and make it more challenging.

### 3 The GSM-Identity Pipeline

This section describes the methodology employed to create a new dynamic benchmark for the evaluation of LLMs with respect to mathematical complexity in high school level mathematics problems.

The key idea behind our approach consists in modifying questions in the GSM8K dataset by representing numbers in a different way, while keeping the same exact meaning. Our approach has a twofold goal: it aims at developing a more challenging benchmark to

<sup>3</sup>Examples from this dataset are too long to be added.

measure mathematical understanding of LLMs, and it tackles the reasoning/memorization dilemma by creating a dynamic dataset that is by design impossible to find in the LLMs training corpus.

Moreover, since we modify existing questions instead of creating new ones, we also ensure that the questions difficulty is in line with previous works and our approach only adds difficulty to otherwise simple questions.

We selected three types of modifications, which we apply to the questions in GSM8K:

1. **BODMAS:** which consists in adding 0 or multiplying by 1 done in non-obvious ways, such as replacing  $8 * 1$  with  $8 * 2/2$  and similar arithmetic operations<sup>4</sup>
2. **Inverse:** which consists in applying a function and its inverse, thereby leaving everything unchanged, we focus on  $\log(\text{exponential}(x))$ ;
3. **Trigonometry:** similar to BODMAS but the number 1 is represented through a Trigonometric Identity such as  $1 = \sin^2(x) + \cos^2(x)$ .

Table 1 reports the Equations used to define each modification. We remark how the modifications we add to the dataset are still within reach for grade school students and we believe that it still holds that, as mentioned by Cobbe et al. (2021), “All the questions should be within reach for a bright middle school student”.

In this work we introduce GSM-Identity, a pipeline to dynamically increase the difficulty of the GSM8K dataset, which is composed of three steps:

1. **Extraction:** Given a question from the original dataset, a regex pattern is used to select an integer from the question.
2. **Replacement:** The integer value is then replaced with a more complex mathematical expression taken from BODMAS, Inverse, or Trigonometry.
3. **Verification:** We evaluate the new mathematical expression to make sure that it evaluates to the same integer value that was extracted by the regex.

**Table 1** The three types of mathematical complexity added to GSM8K

Type	Equation	Example of use
Mathematical operation (+-*/) [BODMAS]	$1 = n/n$	$8 = 7 + (42/42)$
	$1 = 0.5 * 2$	$8 = 8 * 0.5 * 2$
Log-exponential function [inverse]	$x = \log(\text{exponential}(x))$	$8 = \log(\text{exponential}(8))$
	$x = \text{exponential}(\log(x))$	$8 = \text{exponential}(\log(8))$
Trigonometric identities [trigonometry]	$1 = \cos^2(n) + \sin^2(n)$	$8 = 7 + (\cos^2(28) + \sin^2(28))$
	$1 = \sec^2(n) - \tan^2(n)$	$8 = 8/(\sec^2(72) - \tan^2(72))$
	$1 = \text{cosec}^2(n) - \cot^2(n)$	$8 = 9 - (\text{cosec}^2(6) - \cot^2(6))$

The variable  $x$  indicates a number in a GSM8K question. In equations that have an additional variable  $n$ ,  $n$  is replaced with a random natural number from 1 to 100 when applied to a question

<sup>4</sup>We chose BODMAS; as the name because it is a typical acronym used to help children remember the order of arithmetic operations.

Each step is devoted to ensure a different property of the newly developed dataset. The **Extraction** step ensures that we don't always pick the same integer to change when modifying a given question, the **Replacement** step ensures that the new expression is more complex than the original integer value, and the **Verification** step ensures the values in the question are not modified and thus the correct answer is unchanged. While the GSM8K dataset is our main focus, we also test with other datasets (see Sect. 7) and the GSM-Identity pipeline can be extended to other datasets as well with minor adaptations.

Although there are many possible variations to our approach that we could leverage to make GSM-Identity more challenging, in this work we focus on the three modifications in Table 1 and integer values, as this setup proved to be already challenging for some models, as shown in Sect. 4.4. In Table 2 we show an example of a question from GSM8K that has been modified by each of the modifications proposed in Table 1.

Each execution of GSM-Identity creates a novel dataset, sharing the same underlying logic but with different values, so as to counter the possibility of memorization of published data. We still release the version of the dataset on which we test LLMs to ensure the replicability of our results (see details in Sect. 4.3). The implementation of GSM-Identity is available under an open-source license.<sup>5</sup>

## 4 Experiments

Our experiments aim at comparing the accuracy achieved by a selection of LLMs on the GSM8K dataset and the GSM-Identity dataset. We also investigate how the use of prompts with varying level of information about the task impacts the accuracy.

### 4.1 Models

We compare a selection of mid-size and large-size open-source LLMs, representing the recent and current state of the art: Llama (Touvron et al., 2023), Mistral (Mistral AI Team, 2023), Gemma (Team et al., 2024), DeepSeek (2025), and Qwen (Yang et al., 2024). Table

**Table 2** Examples of GSM8K questions modified with mathematical complexity which resolve to the same numeric value

Question type	Example
Original GSM8K	James decides to run 3 sprints 3 times a week. He runs 60 m each sprint. How many total meters does he run a week?
BODMAS complex	James decides to run 3 sprints $(3.0 * (0.5 * 2))$ times a week. He runs 60 m each sprint. How many total meters does he run a week?
Inverse complex	James decides to run 3 sprints $\log(\text{exponential}(3))$ times a week. He runs 60 m each sprint. How many total meters does he run a week?
Trigonometry complex	James decides to run 3 sprints $(2 + (\sec^2(74) - \tan^2(74)))$ times a week. He runs 60 m each sprint. How many total meters does he run a week?

3 presents the detailed list of the tested models, along with the corresponding shortened

<sup>5</sup>[https://github.com/kajalnegi/complex\\_math\\_evaluation](https://github.com/kajalnegi/complex_math_evaluation).

**Table 3** Summary of tested models

Identifier	Number of parameters	Transformer layers	Context length	HuggingFace name
deepseek-llama-8B	8B	32	128k	deepseek-ai/DeepSeek-R1-Distill-Llama-8B
deepseek-math-7B	7B	32	4k	deepseek-ai/deepseekmath-7b-instruct
gemma-2-9B	9B	42	8k	google/gemma-2-9b-it
mistral-nemo	12B	40	128k	mistral-ai/Mistral-Nemo-Instruct-2407
llama-3.1-8B	8B	32	128k	meta-llama/Llama-3.1-8B-Instruct
llama-3.3-70B	70B	80	128k	meta-llama/Llama-3.3-70B-Instruct
qwen2.5-7B	7B	28	4k	Qwen/Qwen2.5-7B-Instruct
qwen2.5-math-1.5B	1.5B	28	4k	Qwen/Qwen2.5-Math-1.5B-Instruct
qwen2.5-math-7B	7B	28	4k	Qwen/Qwen2.5-Math-7B-Instruct
qwen2.5-math-72B	72B	80	4k	Qwen/Qwen2.5-Math-72B-Instruct

The letter “B” indicates billions of parameters

identifiers that will be utilized throughout the paper.<sup>6</sup> We have chosen models from different model families to make the results of our work more robust. In Sect. 4.5 we focus on understanding the impact of model size by studying the performance of qwen2.5-math models with varying parameter counts.

We indicate the model parameters count as a number followed by the letter *B* e.g., *7B* indicates a model with approximately 7 billion parameters, as this has become standard practice when naming models.

## 4.2 Prompts

We compare the use of four different prompts, to investigate how a different prompt affects the mathematical reasoning capabilities of the models.

- **Generic:** This prompt simply presents the question and asks the model to solve it (see Fig. 1a). This is the least informed prompt, which does not give any hint about the eventual presence of mathematical identities in the questions.
- **Notice:** This prompt warns the model that the question has mathematical identities and trivialities, by adding an extra line to the **Generic** prompt: “*Remember, this question contains mathematical identities and trivialities; therefore, resolve them before calculating the answer.*”. This prompt still does not contain explicit information about what to expect.
- **5-CoT:** This prompt contains five examples of questions with their answers, including a Chain of Thought (CoT) (Wei et al., 2022) in the answer of each example. The five examples are sampled from the training part of GSM8K. The CoT reasoning process is available from GSM8K, which consists of a text with step-by-step computation of the

<sup>6</sup>We run all the tested models except qwen2.5-math-1.5B, qwen2.5-math-7B and qwen2.5-math-72B using the vLLM package (Kwon et al., 2023), available at <https://vllm.ai/> which increased by an order of magnitude the inference speed with respect to the commonly used Hugging Face’s transformer package (Wolf et al., 2020), <https://huggingface.co/transformers>, with minimal variations in the output. For the qwen2.5-math-1.5B, qwen2.5-math-7B and qwen2.5-math-72B model, we used Hugging Face’s transformer package, because of an unresolved bug in vLLM at the time of writing this paper, which caused vLLM to produce nearly random text when doing inference with these specific model.

**Generic / Notice**

Here are some mathematical questions along with their answers. Solve the following mathematics problem: {problem.text} Provide your solution in the following format:

1. A step-by-step brief numeric calculations on how to arrive at the solution (No programming code)
2. Place the final numeric answer without any unit or sentence after #####

*Remember that this question contains mathematical identities and trivialities; therefore, resolve them before calculating the answer.*

Also, this is a high school level problem, so advanced mathematical concepts should not be used. Always follow the format.

**5 CoT / Complex 5 CoT**

Here are some mathematical questions along with their answers.

Question 1: {Question} Answer 1: {CoT Solution} ##### {Final Answer}; Question 2: {Question} Answer 2: {CoT Solution} ##### {Final Answer} ... Question 5: {Question} Answer 5: {CoT Solution} ##### {Final Answer}

Solve the following mathematics problem: {problem.text} Provide your solution in the following format:

1. A step-by-step brief numeric calculations on how to arrive at the solution (No programming code)
2. Place the final numeric answer without any unit or sentence after #####

Remember that this question contains mathematical identities and trivialities; therefore, resolve them before calculating the answer.

Also, this is a high school level problem, so advanced mathematical concepts should not be used. Always follow the format.

(a) Generic and Notice prompts. The Generic prompt does not include the *italic red sentence*, which is part of the Notice prompt.

(b) Chain of Thought (CoT) prompts. In **green** the examples provided as context to the model with extra instructions.

**Fig. 1** Prompting strategies

answer for each question. In 5-CoT prompts the examples do not contain the mathematical complexities the actual question may contain. The template of the prompt is shown in Fig. 1b. This prompt is informative with respect to the type of question the model will be asked, yet it does not provide any example with mathematical complexities.

- **Complex 5-CoT:** This prompt follows the same template as 5-CoT shown in Fig. 1b, but the examples contain mathematical complexities from Table 1. For each example, we expand its CoT reasoning process to include the solution of the mathematical complexity in the first step. This is the most informative prompt, giving explicit information about the types of question and the mathematical complexities.

#### 4.3 Datasets

As already mentioned, we experiment with the GSM8K dataset (Cobbe et al., 2021), specifically its test set. The test set contains 1319 samples of high school math questions, along

with their answers and brief calculations to reach those answers. For the given examples and experiments, exactly one integer in each question is replaced. This collection of 1,319 samples serves as the basis for generating the GSM-Identity dataset that incorporates mathematical trivialities and identities. The GSM-Identity dataset thus consists of three sets of 1319 questions, one set for each type of complexity: BODMAS, Inverse, Trigonometry.

Of the 1319 questions in the GSM8K dataset, no new questions were created for 65 of them. This occurs when all the digits in the question are decimal or in word form, for questions like *Jan has three times the number of pets as Marcia. Marcia has two more pets than Cindy. If Cindy has four pets, how many total pets do the three have?*. We kept these 65 questions in both datasets in order to preserve the comparability with other work using GSM8K. Although our pipeline is capable of producing a fresh dataset in each iteration, for reproducibility a frozen dataset is used to generate results for all experiments reported in this manuscript. The frozen dataset is released on GitHub.

The examples used in CoT or Complex CoT come from GSM8K train dataset. The same examples are used for all the prompts. GSM8K and the datasets in Sect. 7 are composed of high-school math problems and are very similar in structure. GSM-Identity adds similar complexity in each dataset. Complex CoT has examples from BODMAS, Inverse and Trigonometry. The frozen examples given are in Tables 10 and 11. The same prompts are used and formatted via “Hugging Face pipeline” for all models. When using these prompting mechanisms, we have observed a reasonable accuracy and low parse error in model responses.

## 4.4 Results

We have tested all the models on both GSM8K and GSM-Identity, using all the prompts, for a total of 128 configurations. The answers are extracted from the model-generated output using a set of parsing patterns to determine the final integer answer. As requested in the prompt, we used “#####” to parse the correct answer. We found that many models follow the request for the output format. To accommodate answers where the model did not accurately follow the requested format, the model answer is parsed using phrases like “Final Answer is” or “Answer”. Listing 1 gives a snippet of code used to parse the model answer. Using this simple policy produced a low parse error, as shown in Table 6.

```
1 def find_answer(row):
2     # list of possible markers for the position of the answer
3     split1= row[column].rsplit('#####', 1)
4     split2 = row[column].rsplit('\boxed(', 1)
5     split3 = row[column].rsplit('**Final Answer:**')
6     split4 = row[column].rsplit('**Answer:**')
7     # code continues....
```

Listing 1: Parse logic for model answer

Table 4 presents the accuracy attained by LLMs when answering GSM8K and GSM-Identity questions, from these results we draw the following observations:

- **Size versus performance relationship:** The 70B parameter models (llama-3.3-70B and qwen2.5-math-72B) achieved near-saturation performance on original GSM8K questions (~ 95% accuracy), maintaining strong results (89–94% accuracy) even with mathematical complexities. Mid-sized models (7-12B parameters) showed more vari-

**Table 4** Accuracy of the models with different prompting for GSM8K questions and GSM-Identity questions

Model	Prompt	GSM8K	BODMAS	Inverse	Trigonometry	Avg
llama-3.1-8B	Generic	71.6	63.8	<u>33.6</u>	42.2	46.5
	Notice	67.5	62.2	42.3	51.3	51.9
	5-CoT	71.6	63.6	41.4	54.3	53.1
	Complex 5-CoT	<b>74.6</b>	65.4	59.1	55.1	59.9
mistral-nemo	Generic	73.1	61.9	<u>33.2</u>	50.2	48.4
	Notice	74.5	63.1	35.3	56.6	51.7
	5-CoT	80.7	70.0	38.7	60.7	56.5
	Complex 5-CoT	<b>83.0</b>	73.4	71.9	68.8	71.4
deepseek-llama-8B	Generic	66.6	57.6	<u>29.2</u>	47.5	44.8
	Notice	69.7	56.9	33.0	61.6	50.5
	5-CoT	<b>72.9</b>	61.3	33.7	61.9	52.3
	Complex 5-CoT	70.1	61.8	59.1	58.8	59.9
deepseek-math-7B	Generic	75.3	64.6	54.6	<u>44.1</u>	54.4
	Notice	<b>75.7</b>	65.4	58.1	53.3	58.9
	5-CoT	71.2	60.0	52.5	48.3	53.6
	Complex 5-CoT	73.7	63.6	64.5	59.5	62.5
gemma-2-9B	Generic	85.8	79.0	<u>35.6</u>	58.0	57.5
	Notice	84.8	78.1	54.4	76.8	69.8
	5-CoT	85.0	76.4	51.2	71.9	66.5
	Complex 5-CoT	<b>86.4</b>	79.8	72.8	77.0	76.5
qwen2.5-math-1.5B	Generic	<b>85.2</b>	81.2	76.2	78.8	78.7
	Notice	84.7	80.7	76.5	77.4	78.2
	5-CoT	80.5	75.6	74.8	74.1	74.8
	Complex 5-CoT	82.0	77.0	<u>56.7</u>	75.3	69.7
qwen2.5-7B	Generic	90.8	88.2	<u>72.6</u>	87.8	82.9
	Notice	91.4	89.2	80.4	87.1	85.6
	5-CoT	<b>91.6</b>	88.2	78.4	86.8	84.5
	Complex 5-CoT	91.0	86.1	87.0	86.1	86.4
qwen2.5-math-7B	Generic	<b>94.6</b>	91.8	89.5	91.2	90.8
	Notice	94.0	92.4	88.6	90.9	90.6
	5-CoT	94.2	89.2	<u>87.1</u>	88.9	88.4
	Complex 5-CoT	93.9	90.0	91.0	89.1	90.0
llama-3.3-70B	Generic	<b>95.8</b>	94.1	<u>89.8</u>	90.8	91.6
	Notice	95.4	94.4	93.9	94.0	94.1
	5-CoT	95.6	94.7	93.8	93.6	94.0
	Complex 5-CoT	95.2	93.9	94.7	93.9	94.2
qwen2.5-math-72B	Generic	95.7	95.1	93.9	94.2	94.4
	Notice	<b>95.8</b>	95.1	94.2	93.8	94.4
	5-CoT	95.5	94.7	94.3	93.5	94.2
	Complex 5-CoT	95.7	94.9	94.4	<u>93.4</u>	94.2

The average is calculated only on GSM-Identity questions. Boldface values are best accuracy and underlined values are worst accuracy across prompts and datasets for each model

ability, going from 46% to 86% accuracy on complex variants.

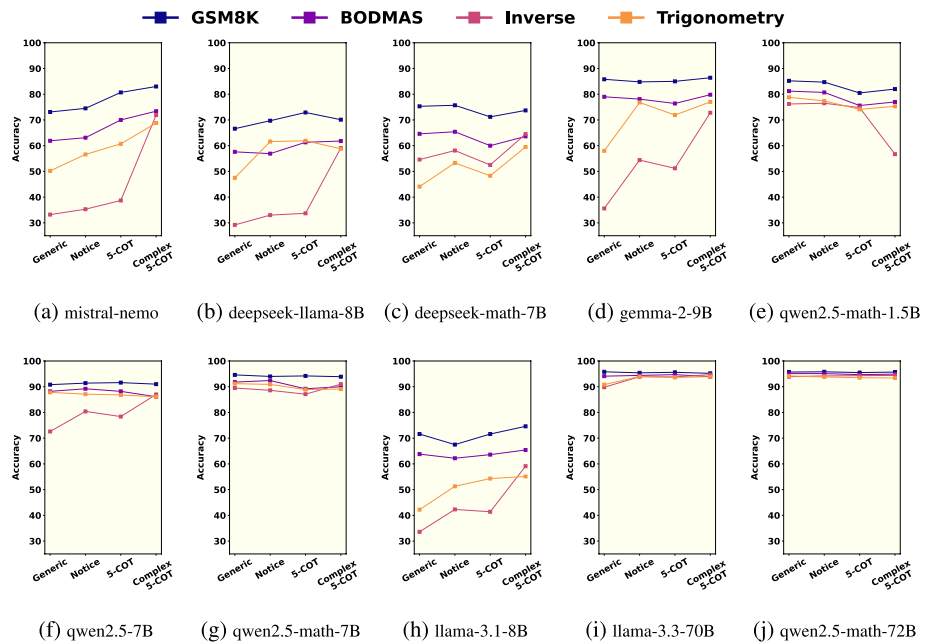
- **Prompting strategy efficacy:** Complex 5-CoT prompt emerged as the top performer in 19 out of 32 evaluations. It also provided the most consistent improvements, with accuracy gain ranging from 3.5% (qwen2.5-72B) to 23% (mistral-nemo), and an aver-

age gain of 13.6% for 7B models. On 70B models, llama3.3-70B had gain of 2.6%. Although qwen2.5-math-72B did not show improvement, it is the only model that demonstrates almost constant accuracy across the various prompts.

- **Architecture differences:** Math-specialized models (deepseek-math-7B, qwen2.5-math-72B) showed a lower weakness on Inverse compared to the other type of complexity. Other models frequently performed worse on Inverse, and had the best benefit from the Complex 5-CoT prompt in solving this type of questions.
- **Performance plateaus:** 70B models appeared near the ceiling effect for original GSM8K questions. However, on complex variants, even top models showed consistent error patterns, with lower accuracy on Inverse and Trigonometry. Moreover, less than half of the errors made by the two models are shared between them, which is a pattern that is replicated also by the 7B models (see Tables 17, 18 and 19 in Appendix), indicating that there is still a substantial set of solvable questions answered incorrectly by either of the models due to their respective limitations.

Figure 2 shows the accuracy trends associated with each model, question type, and prompt type. The figures visualize the superior performance of the 70B models, the difficulties of some models with Inverse, and the trends of 7B models to improve their accuracy as they have more information from the prompt.

We conducted an additional study to assess the effects of adding increasing levels of complexity to math problems, by repeatedly applying GSM-Identity. We have described this process in detail in Appendix A. By applying GSM-Identity up to five times, we observed an additional reduction of 5% in average accuracy as compared to GSM8K for any type of



**Fig. 2** Accuracy of the tested models with respect to question type and prompt format: **a** mistral-nemo, in **b** deepseek-llama-8B, in **c** deepseek-math-7B, in **d** gemma-2-9B, in **e** qwen2.5-math-1.5B, in **f** qwen2.5-7B, in **g** qwen2.5-math-7B, **h** llama-3.1-8B, in **i** llama-3.3-70B and in **j** qwen2.5-math-72B

prompt using the model llama-3.3-70B. The possibility to apply GSM-Identity repeatedly is also part of the GSM-Identity pipeline implementation, so as to allow the generation of datasets of increasing difficulty.

#### 4.5 Model Size Ablation

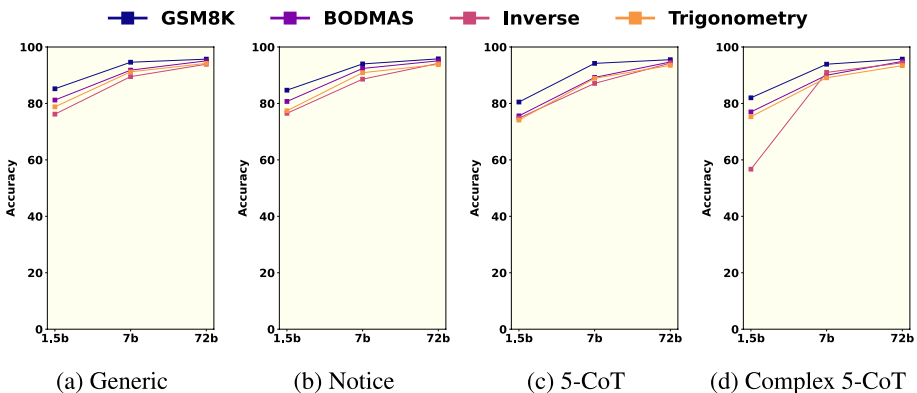
We studied the impact of model size within the same family of Qwen models. For this purpose we used qwen2.5-math-1.5B, qwen2.5-math-7B and qwen2.5-math-72B, as they share the same architecture but vary in size. Figure 3 shows the trends of accuracy of the different prompt with respect to model size and question types. As expected, we observe an increase in accuracy with increasing model size in both GSM8K and GSM-Identity datasets.

A comparison can also be made with the qwen2.5-7B model, which shares the architecture with the qwen2.5-math models but has a different training, as it is not fine tuned with math corpus. As expected, qwen2.5-7B shows lower accuracy in all prompts with GSM8K and GSM-Identity dataset when compared to qwen2.5-math-7B (see Table 4). In a case, i.e., Inverse questions with Generic prompt, qwen2.5-7B (72.6) performs even worse than the qwen2.5-math-1.5B (76.2) model despite its larger size.

### 5 Error Analysis

In this section, we describe the manual annotation we conducted to understand the errors in model answers on the GSM-Identity dataset and to examine whether there are differences between the errors made by mid-scale and large-scale models.

The manual annotation is performed to obtain qualitative and quantitative insights into the model's behavior on the GSM-Identity dataset. We identify four properties that must be satisfied when a model answers the modified questions we propose. As we use mathematical identities to introduce complex expressions in each question, models must identify these expressions, resolve them, determine the correct answer, and output it in the correct format. More in detail:



**Fig. 3** Accuracy trends within qwen2.5-math architecture for GSM8K and GSM-Identity with different prompts, Generic in (a), Notice in (b), 5-CoT in (c), Complex 5-CoT in (d)

- (a) **Identification of complexity:** When answering the question, the model should identify the complex expression we introduced in the GSM8K question.
- (b) **Resolution of complexity:** The model should resolve the identified complex expression in order to recover the original numerical value.
- (c) **Correctness of the final answer:** The model should correctly answer the question and match the final numeric answer.
- (d) **Correctness of the answer pattern:** The model should adhere to the answer pattern requested even if the question has the additional expression introduced in GSM-Identity.

To determine whether a model's answer exhibits each desired property, we manually annotated each property independently:

- (a) We check whether the model identifies the presence of a complex expression by explicitly mentioning the *complete* complex expression in its answer; otherwise, we assign an error under the "Identification of complexity" label.
- (b) We check whether the model can simplify the identified complex expression to the original numerical value by explicitly stating or evaluating the *equality* of the two in its answer; otherwise, we assign an error under the "Resolution of complexity" label.
- (c) We check the final answer and verify its equality with the *original answer*; if the model fails to provide the correct result, we assign an error under the "Correctness of the final answer" label.
- (d) We check whether the answering pattern is correct and whether the numeric answer can be *extracted* by our parsing logic. If our parser fails to capture it, we assign an error under the "Correctness of the answer pattern" label.

An answer can be annotated with errors in one, or more than one of the properties, making the annotation process a multi-label classification task. To improve clarity of the annotation procedure, Table 5 provides insights into how the manual evaluation has been carried out and it illustrates an example for each label assigned, possible multiple ones for each example, highlighting the incorrect segments in the model's response. In Appendix C, we present additional examples.

To perform comprehensive manual testing while limiting the annotation costs, we annotate the answers of two mid-size and two large-size models. For mid-size, we manually annotate the outcomes of deepseek-llama-8B and gemma-2-9B, the first because it has the lowest accuracy among all the models we tested and the second because it performed low only in Inverse. For large-size models, we annotate llama-3.3-70B and qwen2.5-math-72B, the only models with more than seventy billions parameters among those we studied. To select the answers to annotate, we collected a set of 100 incorrect answers for each of the four models.

This analysis is conducted for the "Notice" prompt within the GSM-Identity dataset. Given the available human annotation resources we had, we chose to examine the model's behavior using the "Notice" prompt, as it gives the models the minimal instruction specific for the GSM-Identity dataset. We prefer "Notice" over both CoT prompts because we observed that CoT and Complex CoT prompts yield improved accuracy (see Table 4). The choice of "Notice" prompt provides insight into the model's behavior in a basic setting for complex math problems. We leave the analysis of more complex prompts to future work.

**Table 5** Examples of the possible type of errors made by a model when prompted as in Fig. 1 with a question (top of the table) that includes a [math complex problem](#)

a) Identification of complexity	b) Resolution of complexity
<p><u>Question:</u>Elise has been selling her Dad's collection of 250 books for three years. Each book sells at 20\$, and she sold twice as many books in the first year as she has sold in the current year. There are currently <math>(51 - (\sec^2(19) - \tan^2(19)))</math> unsold books, and her sales number this year is 45. What's the total amount of money she earned in the second year?</p> <p><u>Answer[<a href="#">gemma-2-9B</a>]:</u> 1. * Calculate the number of books sold in the first year: 45 books * 2 = 90 books * Calculate the number of books sold in the second year: 90 books - 45 books = 45 books * Calculate the total earnings in the second year: 45 books * \$20/book = \$900 #### 900</p> <p><u>Annotation:</u>  <ul style="list-style-type: none"> <li>✗ Recognised complexity</li> <li>✗ Resolved complexity</li> <li>✗ Arrived at correct answer</li> <li>✓ Correct answer pattern</li> </ul> </p>	<p><u>Question:</u> There are 220 castles in Scotland. <a href="#">exponential(log(40))</a> percent of them are ruins, and half of the ruined castles are unmanned. How many unmanned ruined castles are there in Scotland?</p> <p><u>Answer[<a href="#">gemma-2-9B</a>]:</u> 1. * Calculate log(40) * <a href="#">Calculate exponential(log(40))</a> * Multiply the result from step 2 by 220 to find the number of ruined castles. * Divide the result from step 3 by 2 to find the number of unmanned ruined castles. #### 110</p> <p><u>Annotation:</u>  <ul style="list-style-type: none"> <li>✓ Recognised complexity</li> <li>✗ Resolved complexity</li> <li>✗ Arrived at correct answer</li> <li>✓ Correct answer pattern</li> </ul> </p>
c) Correctness of the final answer	d) Correctness of the answer pattern
<p><u>Question:</u>Mico and Marco wanted to get to know each other. They realized that the sum of their ages is 20. What will be the sum of their ages in <a href="#">(10.0*(15/15))</a> years?</p> <p><u>Answer[<a href="#">gemma-2-9B</a>]:</u> 1. <math>10.0 * (15/15) = 10.0 * 1 = 10</math> 2. <math>20 + 10 = 30</math> #### 30</p> <p><u>Annotation:</u>  <ul style="list-style-type: none"> <li>✓ Recognised complexity</li> <li>✓ Resolved complexity</li> <li>✗ Arrived at correct answer</li> <li>✓ Correct answer pattern</li> </ul> </p>	<p><u>Question:</u> Adrian's age is three times the age of Harriet, and Harriet is half the age of Zack. Calculate the average age of the three in three years if Harriet is <a href="#">log(exponential(21))</a> years old now.</p> <p><u>Answer[<a href="#">gemma-2-9B</a>]:</u> 1. Harriet's age now = <math>\log(\text{exponential}(21)) = 21</math> years (Since log and exponential are inverse functions) Adrian's age now = <math>3 * \text{Harriet's age} = 3 * 21 = 63</math> years Zack's age now = <math>2 * \text{Harriet's age} = 2 * 21 = 42</math> years Harriet's age in three years = <math>21 + 3 = 24</math> years Adrian's age in three years = <math>63 + 3 = 66</math> years Zack's age in three years = <math>42 + 3 = 45</math> years Average age in three years = <math>(\text{Harriet's age} + \text{Adrian's age} + \text{Zack's age}) / 3 = (24 + 66 + 45) / 3</math> <u>2. 45</u></p> <p><u>Annotation:</u>  <ul style="list-style-type: none"> <li>✓ Recognised complexity</li> <li>✓ Resolved complexity</li> <li>✓ Arrived at correct answer</li> <li>✗ Correct answer pattern</li> </ul> </p>

Text in the [Annotation](#) section gives a step by step evaluation of the stages passed and the error. [Text in orange in the Answer is the failed stage of evaluation](#) [Text in green in the Answer is the intended final answer by the model.](#)

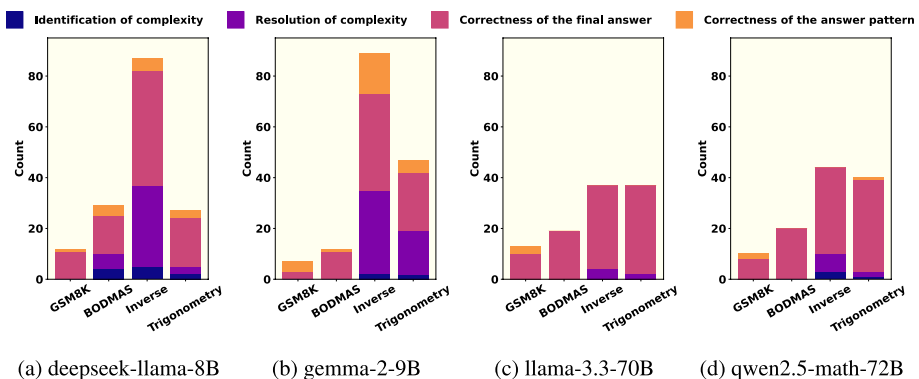
The annotation was performed by two annotators, both holding an MSc degree in Mathematics, yielding an average inter-annotator agreement of 0.69 as measured by Cohen's kappa coefficient (McHugh, 2012). The expected agreement when both annotators assign labels randomly is estimated using a per-annotator empirical prior over the class labels. We compute Cohen's Kappa coefficient in the multilabel setup, computing a separate score for each error category. The resulting kappa values are 0.12 for "Identification of complexity", 0.84 for "Resolution of complexity", 1.0 for "Correctness of the final answer", and 0.79 for "Correctness of the answer pattern". The kappa value is significantly low for "Identification of complexity" and above 0.79 for other categories.

Across all four models and for every dataset type, the most occurring error category is "Correctness of the final answer", where the final computed result does not match the correct numeric answer, as shown in Fig. 4. "Identification of complexity" has a very low count, indicating that all four models acknowledge the presence of the added complexity. "Resolution of complexity" is the second most common error type for deepseek-llama-8B and gemma-2-9B for Inverse, which is a key difference between small and large models, as larger models make very few errors of this type.

The Inverse complexity is the one that caused more errors for two models deepseek-llama-8B and gemma-2-9B, as shown in Table 6. For llama-3.3-70B and qwen2.5-math-72B, Trigonometry has the highest count of errors in the category "Correctness of the final answer".

Interestingly, we find that in 13 out of the 31 errors of this type for gemma-2-9B, the returned answer was correct if simplified to an integer value. Overall, large models exhibit a significantly lower count of "Resolution of complexity" errors compared to mid-sized models. Only gemma-2-9B exhibited a relatively high count for "Correctness of the answer pattern" for Inverse dataset. We adopt a multi-label classification setup, allowing each sample to belong to multiple classes; consequently, the error counts for all models exceed the annotated sample size (100). Out of 400 annotated samples, 360 were annotated under the class "Correctness of the answer pattern".

Large-size models show different patterns in the type of errors. They have no "Identification of complexity" for BODMAS and low errors of type "Resolution of complexity". Overall, most of the errors committed by large-size models are "Correctness of the final



**Fig. 4** Error count for 4 types of annotation class by models: in **a** deepseek-math-7B, in **b** gemma-2-9B, in **c** llama-3.3-70B and in **d** qwen2.5-math-72B

**Table 6** Counts of each annotation error class for deepseek-llama-8B, gemma-2-9B, llama-3.3-70B, and qwen2.5-math-72B using the prompt “*Notice*”

	Type of error				Total errors	
	Identification of complexity	Resolution of complexity	Correctness of the final answer	Correctness of the answer pattern		
deepseek-llama-8B	No Complexity (GSM8K)	0	0	11	1	12
	BODMAS	4	6	15	4	29
	Inverse	5	32	45	5	87
gemma-2-9B	Trigonometry	2	3	19	3	27
	No Complexity (GSM8K)	0	0	3	4	7
	BODMAS	0	0	11	1	12
	Inverse	2	33	38	16	89
llama-3.3-70B	Trigonometry	2	17	23	5	47
	No Complexity (GSM8K)	0	0	10	3	13
	BODMAS	0	0	19	0	19
	Inverse	0	4	33	0	37
qwen2.5-math-72B	Trigonometry	0	2	35	0	37
	No Complexity (GSM8K)	0	0	8	2	10
	BODMAS	0	0	20	0	20
	Inverse	3	7	34	0	44
Total errors	Trigonometry	1	2	36	1	40
		19	106	360	45	

Rows labeled “No Complexity (GSM8K)” contain samples from GSM8K that are not transformed by GSM-Identity. **Bold face** indicates the highest value in each row, while underlining indicates the lowest error count

answer”, indicating that the added complexity is recognized and resolved, but it still makes it harder for the model to produce the correct answer.

## 6 Comparing with Humans

We conducted three human evaluations using the Prolific crowdsourcing platform. The purpose of our human study is to understand how challenging GSM-Identity is for humans and if there are similarities between humans and models when resolving the complexities in our dataset.

For all experiments, participants were required to have English as their native language and to possess an educational qualification of at least a high-school diploma. It facilitates a comprehension of both questioning techniques and mathematical terminology. In any study reported in this section, a participant was presented with a batch of 5 questions and was paid 9£/hour for the participation in the study. Participants took a median time of 5 min to answer the questions in a batch.

### 6.1 Humans Compare GSM-Identity to GSM8K for Equivalence

Our first human study evaluates the ability of the annotators to assess the equivalence of mathematical expressions. In particular, we want to answer the question: Can humans resolve the complexities added in GSM-Identity? That is, we want to understand if annotators can tell if a question modified by GSM-Identity is equivalent to the original one. We perform the test by displaying two questions to users. In half of the questions, one of the numerical values in the question is replaced with a GSM-Identity expression equivalent to the original value, in the other half the GSM-Identity expression evaluates to a different value. Users are expected to answer whether the two questions are mathematically equivalent or not, as shown in Fig. 5a.

The human study consisted of 100 equivalent and non-equivalent pairs of questions (created from the same 100 original questions). Overall, prolific users had an accuracy of 73%. This confirms that prolific users answered questions without randomly selecting an answer, and provides us with a reference for the study in which we ask humans to solve the questions giving a numerical answer (see Sect. 6.3).

With respect to the human accuracy for each type of complexity, on BODMAS humans have the highest accuracy, 77.4%, then Inverse, 72.7%, and Trigonometry is the lowest, 68.0%.

Therefore, humans can resolve the complexities added in GSM-Identity, as the annotators perform above random chance and with an accuracy of more than 70% in both BODMAS and Inverse.

### 6.2 Human and Model Errors

We are also interested in understanding if humans and models face similar difficulties when exposed to GSM-Identity questions. In particular, we want to answer the question: Do humans and models fail to recognize the same mathematical complexities? To measure this, we replicate the equivalence study using Fig. 5a, but we select examples where models

**Equivalence test**

Question 1:  
A candle melts by 2 centimeters every hour that it burns. How many centimeters shorter will a candle be after burning from 1:00 PM to 5:00 PM?

Question 2:  
A candle melts by  $\log(\text{exponential}(2))$  centimeters every hour that it burns. How many centimeters shorter will a candle be after burning from 1:00 PM to 5:00 PM?

Select an answer:  
 Question 1 and Question 2 are equivalent  
 Question 1 and Question 2 are not equivalent

**Answer test**

In this study you will be presented with mathematical questions, requiring a natural number as the answer.

Questions are of the kind: “Mico and Marco wanted to get to know each other. They realized that the sum of their ages is 20. What will be the sum of their ages in 10 years?”

Some of the questions have a “numerical trick” that changes the representation of a number.

For example the number 8 can be written as  $9 - 11/11$ , or as  $\text{exponential}(\log(8))$ , or  $7 + \sin^2(10) + \cos^2(10)$ .

In this case you have to convert the trick into a natural number and then answer the question.

This study is about evaluating the impact of the tricks on the accuracy of the answers by humans.

It will give us a reference for a similar study on AI models.

For this reason we ask you to not use ChatGPT or similar AI tools to get the answer.

However, feel free to use a calculator for the basic operations.

(a) Example question for the Equivalence test given to Prolific users.

(b) Instructions given to Prolific users for the Answer test to GSM8k and GSM-Identity questions.

**Fig. 5** Human Study Questions

**Table 7** Accuracy of humans on the equivalence test for the samples of the manual error analysis

Human accuracy	
<i>deepseek-llama-8B Errors</i>	
BODMAS	82.4
Inverse	88.0
Trigonometry	66.6
<i>gemma-2-9B Errors</i>	
BODMAS	91.6
Inverse	90.6
Trigonometry	60.7

failed to give the correct answer. We use GSM8K and GSM-Identity questions contributing to positive values in Table 6 and provide them to the annotators to assess the equivalence between the original questions and their corresponding GSM-Identity questions. Table 7 shows the output of this equivalence test.

Interestingly, we find that for BODMAS and Inverse the accuracy of humans in spotting the equivalence of the questions is higher in samples that models could not answer correctly. For Trigonometry, the accuracy was poorer in both studies. Therefore we infer that humans do not show the same level of accuracy in resolving the questions where model failed to answer. Humans are less affected by sample selection and find Trigonometry hardest to resolve consecutively.

### 6.3 Humans Answer the GSM8K and GSM-Identity Questions

In this last human evaluation, we are interested in evaluating how hard are the questions in GSM-Identity for humans, specifically we want to answer the following question: Can humans answer the questions in GSM-Identity? To measure this, we ask annotators to answer the four versions of the questions we have, i.e.: the original version of the GSM8K dataset and three versions from GSM-Identity. We randomly sampled 50 GSM8K questions and made the corresponding versions of BODMAS, Inverse and Trigonometry. The instructions given to human evaluators are presented in Fig. 5b. This study allows us to directly compare the performances of all the models with those of humans. Table 8 shows the accuracy of humans and models on the same set of questions.

Compared to the equivalent test, in the answer test there is an observable drop of 22%, 24% and 38% in accuracy from GSM8K to BODMAS, Inverse and Trigonometry respectively. The trends seen in Sects. 6.1 and 6.2 of humans for the different complexity types are confirmed by this test: Trigonometry is the hardest while BODMAS and Inverse are similar in difficulty. Humans have an accuracy close to the smaller models, and lower than larger models, both on GSM8K and GSM-Identity. A relevant difference is the worst performance, which is in Inverse questions for smaller models and Trigonometry questions for humans.

Human experiment shows that on average there is a drop of 28% in accuracy compared to the original GSM8K, which may be in part ascribed to the uncommon, less familiar aspect of the questions with the added complexity. Yet, it is important to note that the added complexity by GSM-Identity is within high school mathematics knowledge, and limited to a single instance of a number in the question. We also acknowledge that llama-3.3-70B and qwen2.5-math-72B as shown in Table 4, exhibit a negligible or low drop in the accuracy from the GSM8K to GSM-Identity datasets, which justifies their ability to perform well for unseen data. Therefore, models including those intended for educational or tutoring purposes, can benefit from our benchmark as a means of testing their reasoning capabilities and their ability to resolve unseen questions at scale, thereby reducing the risk of mathematical errors while sharing knowledge.

## 7 Additional Datasets

We tested on additional datasets the two top-performing models for both the 7B and 70B sizes, i.e., gemma-2-9B, qwen2.5-7B, llama-3.3-70B, and qwen2.5-math-72B. The additional datasets are part of the Platinum Benchmark (Vendrow et al., 2025), a collection of benchmarks that are carefully curated to minimize label errors and ambiguity, improving the reliability of the evaluation of models.

We selected three datasets from the Platinum Benchmark collection, specifically:

**Table 8** Accuracy of humans in solving the math problems and of the models with different prompting for GSM8K questions and GSM-Identity questions, on a random sample of 50 questions

Model	Prompt	GSM8K	BODMAS	Inverse	Trigonometry	Avg.
humans	Notice with example (see Fig. 5b)	<b>84.0</b>	62.0	60.0	<u>46.0</u>	56.0
llama-3.1-8B	Generic	84.0	82.0	<u>26.0</u>	36.0	48.0
	Notice	78.0	78.0	42.0	54.0	58.0
	5-CoT	<b>86.0</b>	72.0	52.0	56.0	60.0
mistral-nemo	Complex 5-CoT	<b>86.0</b>	74.0	72.0	56.0	67.3
	Generic	80.0	66.0	34.0	58.0	52.7
	Notice	88.0	76.0	<u>26.0</u>	60.0	54.0
deepseek-llama-8B	5-CoT	94.0	70.0	44.0	74.0	62.7
	Complex 5-CoT	<b>96.0</b>	80.0	78.0	74.0	77.3
	Generic	72.0	56.0	<u>32.0</u>	58.0	48.7
deepseek-math-7B	Notice	72.0	66.0	44.0	48.0	52.7
	5-CoT	<b>82.0</b>	72.0	34.0	54.0	53.3
	Complex 5-CoT	<b>82.0</b>	76.0	70.0	58.0	68.0
gemma-2-9B	Generic	<b>86.0</b>	74.0	48.0	<u>46.0</u>	56.0
	Notice	82.0	72.0	60.0	56.0	62.7
	5-CoT	82.0	64.0	62.0	52.0	59.3
qwen2.5-7B	Complex 5-CoT	78.0	66.0	62.0	58.0	62.0
	Generic	96.0	86.0	<u>30.0</u>	54.0	56.7
	Notice	<b>98.0</b>	82.0	54.0	86.0	74.0
llama-3.3-70B	5-CoT	96.0	76.0	74.0	80.0	76.7
	Complex 5-CoT	<b>98.0</b>	88.0	82.0	84.0	84.7
	Generic	96.0	94.0	<u>74.0</u>	94.0	87.3
qwen2.5-72B	Notice	<b>98.0</b>	96.0	90.0	92.0	92.7
	5-CoT	96.0	<b>98.0</b>	84.0	94.0	92.0
	Complex 5-CoT	<b>98.0</b>	92.0	94.0	94.0	93.3
llama-3.3-70B	Generic	<b>98.0</b>	96.0	<u>80.0</u>	88.0	88.0
	Notice	<b>98.0</b>	96.0	94.0	<b>98.0</b>	96.0
	5-CoT	<b>98.0</b>	96.0	94.0	<b>98.0</b>	96.0
qwen2.5-math-72B	Complex 5-CoT	<b>98.0</b>	96.0	96.0	96.0	96.0
	Generic	<b>100.0</b>	96.0	98.0	96.0	96.7
	Notice	<b>100.0</b>	98.0	98.0	96.0	97.3
qwen2.5-math-72B	5-CoT	96.0	<u>92.0</u>	96.0	96.0	94.7
	Complex 5-CoT	98.0	94.0	94.0	<u>92.0</u>	93.3

The average is calculated only on GSM-Identity questions. Boldface values are best accuracy and underlined values are worst accuracy across prompts and datasets for each model

- *multiarith-platinum*: This dataset is composed of questions that are high school level arithmetic problems. It contains 174 questions, which were transformed into math complex questions using GSM-Identity.
- *singleq-platinum*: This dataset is composed of 100 questions which contain high school math problems.
- *svamp-platinum*: Svamp is a dataset of 268 high school math questions which have a body and question field. We prompted models after concatenating the two fields.

In the formulation of CoT-based prompts we sampled the five examples from the same GSM8K training set used for the GSM8K questions.

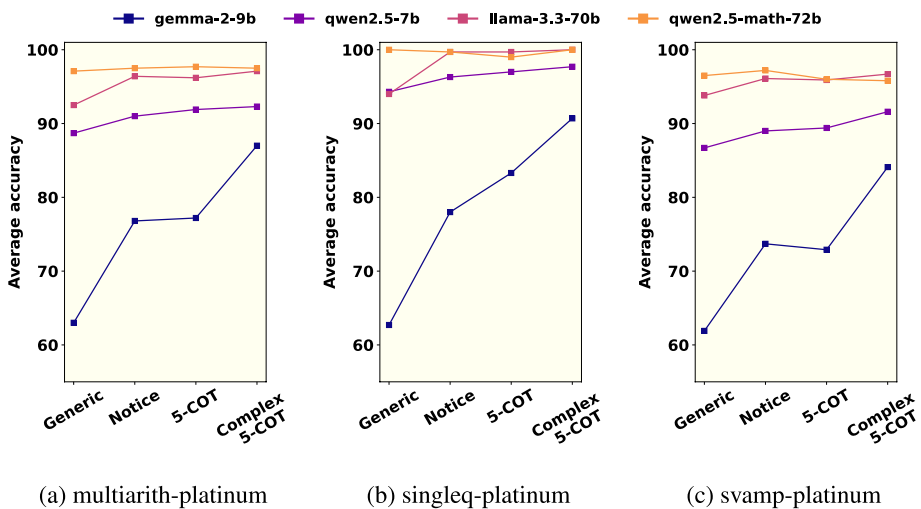
Figure 6 shows the accuracy values obtained by the four models on those datasets (see Table 20 in Appendix for the detailed results on the 192 tested configurations). The accuracy values are all sensibly higher than those based on GSM8K, indicating that GSM8K is the hardest dataset. The Complex 5-CoT prompt is the one that achieves the highest accuracy in 45 case out of 60 (with 18 ties). The model qwen2.5-math-72B demonstrates superior performance in both the original and GSM-Identity datasets. A lower accuracy for Inverse is observed, similar to the trends found in the GSM8K dataset. The larger models, qwen2.5-math-72B and llama-3.3-70B, outperform their mid-sized counterparts in a manner similar to the performance observed on GSM8K.

## 8 Conclusions

In this work, we introduced a novel method for creating more challenging versions of mathematical reasoning benchmarks while maintaining the same conceptual difficulty. By replacing numerical values with equivalent but more complex mathematical expressions in the GSM8K dataset, we developed a framework that can continuously generate new test instances, helping mitigate concerns about data leakage and memorization in LLM evaluation.

We ran experiments on models ranging from 7B to 72B parameters identifying some key insights. While 70B larger models generally outperformed their smaller counterparts, when they are compared to math-oriented 7B models the performance is limited. This suggests that current 7B models represent a promising balance between performance and practical deployability when applied to a specific use-case.

Our human evaluation results provided context for interpreting model performance, revealing that current 7B models achieve accuracy comparable to humans on these modi-



**Fig. 6** Average accuracy of 4 models with respect prompt format: **a** multiarith-platinum, in **b** singleq-platinum, in **c** svamp-platinum

fied mathematical reasoning tasks, while larger models significantly surpass human performance. We also found that different types of mathematical complexity affected models and humans differently—while LLMs struggled most with Inverse function pairs, human participants found Trigonometry identities most challenging. This suggests that while LLMs have made remarkable progress in mathematical reasoning, there remains room for improvement, particularly in handling certain types of mathematical complexity.

The dynamic nature of our approach, where questions can be automatically regenerated with different complex expressions, addresses a critical limitation of static benchmarks that may become memorized by future models.

A promising path for future works includes extending the approach with more complex modifications and possibly combining with approaches proposed in other works.

The techniques and findings presented here contribute to the broader goal of developing more reliable methods for assessing true reasoning capabilities in language models, moving beyond superficial benchmarks that may be susceptible to memorization effects. As LLMs continue to advance, such rigorous evaluation methodologies will become increasingly important for understanding and steering their development.

## 8.1 Limitations

In this section, we highlight the major limitations of our work. GSM-Identity is limited to high-school arithmetic problems while introducing a level of complexity that remains within the scope of high-school mathematics. Mathematics is a varied discipline, widely studied and taught; however, only limited topics are tested in our study. We have seen a decrease in human accuracy from the original GSM8K to GSM-Identity in Sect. 6, making it potentially less useful for human use. This limits our work to research evaluation of LLMs, rather than for real-world purposes or those often taught in educational institutions. The human study was conducted on the crowdsourcing platform Prolific, the results may vary for different individuals or groups.

## Appendix A: Increasing Complexity Level

In our experiment we found that the setup we used for GSM-Identity is a good challenge for 7B-sized models, while models 70B parameter-range obtained very high accuracy value, indicating that GSM-Identity may be considered not challenging for them.

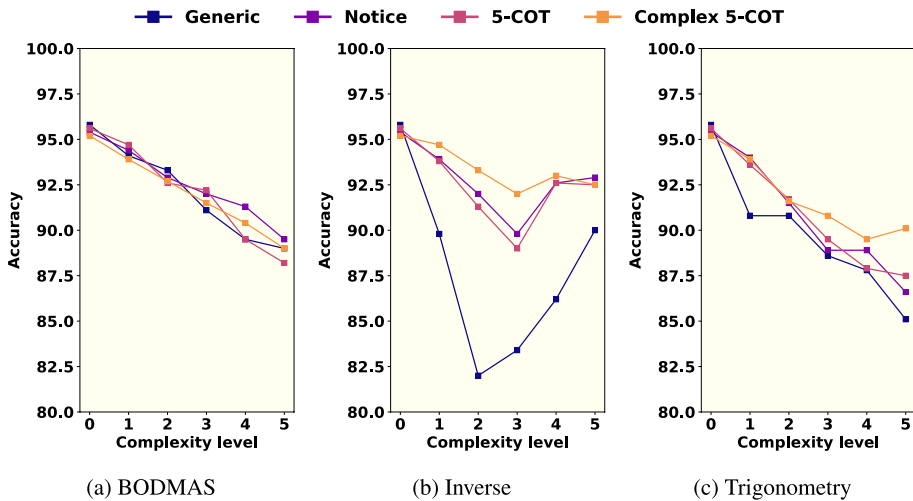
We thus explore how a simple repeated application of our method can increase the difficulty of GSM-Identity. Given a level of complexity  $l \in [1, 2, \dots]$ , for the BODMAS and Trigonometry, we separate the integer into  $l$  components, then we apply the complex modification to each component. For the Inverse type, we nest the Inverse modification  $l$  times. Table 9 presents samples of output for the increased complexity levels.

We evaluated the output of a top-performing model, llama-3.3-70B, and analyzed the trend of accuracy associated with different prompting mechanisms as shown in Fig. 7. In the case of BODMAS and Trigonometry, the models exhibit a steady decline in accuracy as the complexity level increases. When the  $l$  is increased for Inverse, the models demonstrate an ability to recognize the pattern, resulting in either an improvement or a stabilization of accuracy. E.g., in a typical generated response, the llama-3.3-70B model writes “... *this expres-*

**Table 9** Examples of increased complexity in GSM8K reproducing example in Table 2

Question type	Level	Example
Original GSM8K	0	James decides to run 3 sprints 3 times a week. He runs 60 meters each sprint. How many total meters does he run a week?
BODMAS	1	James decides to run 3 sprints $(3.0 * (0.5 * 2))$ times a week. He runs 60 meters each sprint. How many total meters does he run a week?
	2	James decides to run 3 sprints 3 times a week. He runs $(31 - (29/29)) + (30.0 * (64/64))$ meters each sprint. How many total meters does he run a week?
Inverse	1	James decides to run 3 sprints $\log(\text{exponential}(3))$ times a week. He runs 60 meters each sprint. How many total meters does he run a week?
	2	James decides to run 3 sprints 3 times a week. He runs $\text{exponential}(\log(\text{exponential}(\log(60))))$ meters each sprint. How many total meters does he run a week?
Trigonometry	1	James decides to run 3 sprints $(2 + (\sec^2(74) - \tan^2(74)))$ times a week. He runs 60 meters each sprint. How many total meters does he run a week?
	2	James decides to run 3 sprints 3 times a week. He runs $(30.0 * (\text{cosec}^2(5) - \cot^2(5))) + (31 - (\text{cosec}^2(52) - \cot^2(52)))$ meters each sprint. How many total meters does he run a week?

sion simplifies to  $\log(\text{exponential}(\dots))$  repeated several times, which ultimately simplifies to the original number ...” (Table 16 in Appendix illustrates this behavior with examples). By level 5, there is a minimum reduction of 5% in average accuracy as compared to GSM8K for any type of prompt using the model llama-3.3-70B as shown in Table 21.



**Fig. 7** Accuracy trends with increasing complexity level for the model llama-3.3-70B and different prompts, BODMAS in **a** Inverse in **b** Trigonometry in **(c)**

## Appendix B: Examples Used in CoT and Complex CoT

This section contains example from train dataset of GSM8K used in 5-CoT and 5-Complex CoT prompts for all models. Table 10 contains original question and answer from train dataset of GSM8K. Table 11 contains GSM-Identity question derived from train dataset of GSM8K and verified answers.

**Table 10** CoT examples

---

<b>Question 1:</b>	Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?
<b>Answer 1:</b>	Natalia sold $48/2 = \ll 48/2 = 24 \gg 24$ clips in May. Natalia sold $48 + 24 = \ll 48 + 24 = 72 \gg 72$ clips altogether in April and May. ##### 72
<b>Question 2:</b>	Weng earns \$12 an hour for babysitting. Yesterday, she just did 50 min of babysitting. How much did she earn?
<b>Answer 2:</b>	Weng earns $12/60 = \ll \$12/60 = 0.2 \gg 0.2$ per minute. Working 50 min, she earned $0.2 \times 50 = \$\ll 0.2 * 50 = 10 \gg 10$ . ##### 10
<b>Question 3:</b>	Betty is saving money for a new wallet which costs \$100. Betty has only half of the money she needs. Her parents decided to give her \$ 15 for that purpose, and her grandparents twice as much as her parents. How much more money does Betty need to buy the wallet?
<b>Answer 3:</b>	In the beginning, Betty has only $100/2 = \$\ll 100/2 = 50 \gg 50$ . Betty's grandparents gave her $15 * 2 = \$\ll 15 * 2 = 30 \gg 30$ . This means, Betty needs $100 - 50 - 30 - 15 = \$\ll 100 - 50 - 30 - 15 = 5 \gg 5$ more. ##### 5
<b>Question 4:</b>	Julie is reading a 120-page book. Yesterday, she was able to read 12 pages and today, she read twice as many pages as yesterday. If she wants to read half of the remaining pages tomorrow, how many pages should she read?
<b>Answer 4:</b>	Maila read $12 \times 2 = \ll 12 * 2 = 24 \gg 24$ pages today. So she was able to read a total of $12 + 24 = \ll 12 + 24 = 36 \gg 36$ pages since yesterday. There are $120 - 36 = \ll 120 - 36 = 84 \gg 84$ pages left to be read. Since she wants to read half of the remaining pages tomorrow, then she should read $84/2 = \ll 84/2 = 42 \gg 42$ pages. ##### 42
<b>Question 5:</b>	James writes a 3-page letter to 2 different friends twice a week. How many pages does he write a year?
<b>Answer 5:</b>	He writes each friend $3 * 2 = \ll 3 * 2 = 6 \gg 6$ pages a week So he writes $6 * 2 = \ll 6 * 2 = 12 \gg 12$ pages every week That means he writes $12 * 52 = \ll 12 * 52 = 624 \gg 624$ pages a year ##### 624

---

**Table 11** Complex CoT examples

---

<b>Question 1:</b>	Natalia sold clips to $\log(\text{exponential}(48))$ of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?
<b>Answer 1:</b>	* Calculate $\log(\text{exponential}(48))$ : Since the exponential function and the logarithm function are inverses of each other, $\log(\text{exponential}(48)) = 48$ . * Calculate the number of clips sold in May: $48/2 = 24$ * Calculate the total number of clips sold: $48 + 24 = 72$ ##### 72
<b>Question 2:</b>	Weng earns \$12 an hour for babysitting. Yesterday, she just did $(50/(\cos^2(2) + \sin^2(2)))$ minutes of babysitting. How much did she earn?
<b>Answer 2:</b>	* $\cos^2(2) + \sin^2(2) = 1$ (Trigonometric Identity) * $50/(\cos^2(2) + \sin^2(2)) = 50/1 = 50\text{min}$ * Weng earns \$12 per hour, and there are 60 min in an hour. * Earnings = $(50 \text{ min}/60 \text{ min/hour}) * \$12/\text{hour}$ ##### 10
<b>Question 3:</b>	Randy has $(60.0*(0.5*2))$ mango trees on his farm. He also has 5 less than half as many coconut trees as mango trees. How many trees does Randy have in all on his farm?
<b>Answer 3:</b>	* Calculate the number of mango trees: $60.0 * (0.5 * 2) = 60.0$ * Calculate half the number of mango trees: $60.0/2 = 30.0$ * Calculate the number of coconut trees: $30.0 - 5 = 25.0$ * Calculate the total number of trees: $60.0 + 25.0 = 85.0$ ##### 85
<b>Question 4:</b>	A deep-sea monster rises from the waters once every hundred years to feast on a ship and sate its hunger. Over three hundred years, it has consumed $(846+(7/7))$ people. Ships have been built larger over time, so each new ship has twice as many people as the last ship. How many people were on the ship the monster ate in the first hundred years?
<b>Answer 4:</b>	Calculate the total number of people eaten: $846 + (7/7) = 846 + 1 = 847$ people 2. Let 'x' be the number of people on the first ship. 3. The second ship had $2*x$ people. 4. The third ship had $2*(2*x) = 4*x$ people. 5. The total number of people eaten is $x + 2x + 4x = 847$ 6. Combine like terms: $7x = 847$ 7. Divide both sides by 7: $x = 121$ ##### 121
<b>Question 5:</b>	Joy can read 8 pages of a book in 20 min. How many hours will it take her to read $\log(\text{exponential}(120))$ pages?
<b>Answer 5:</b>	**Find Joy's reading rate:** *Joy reads 8 pages/20 min = 0.4 pages per minute. **Calculate the total number of pages: ** * $\log(\text{exponential}(120)) = 120$ (The logarithm and exponential functions cancel each other out) **Calculate the total time in minutes: *** $120 \text{ pages}/0.4 \text{ pages/minute} = 300 \text{ min}$ **Convert minutes to hours: ** * $300 \text{ min}/60 \text{ min/hour} = 5 \text{ h}$ ##### 5

---

## Appendix C: Examples of LLMs Errors

This section reports examples annotated during the error analysis in Sect. 5. Specifically, Table Table 12 shows examples of errors labeled as “Identification of complexity”, Table 13 shows examples of errors labeled as “Resolution of Complexity”, Table 14 shows errors labeled as “Correctness of the final answer” and finally Table 15 shows errors labeled as “Correctness of the Answer Pattern”.

**Table 12** Example of error labeled during manual annotation as “Identification of complexity”

Question	Janeth borrowed \$2000 and promised to return it with an additional 10% of the amount. If she is going to pay \$165 a month for $(12/(0.5 * 2))$ months, how much will be Janeth’s remaining balance by then? [ <b>Correct Answer:</b> 220]
<b>deepseek-llama-8B</b> [Answered correctly in original question without math complexity]	<p>&lt;think&gt; First, I recognize that Janeth borrowed \$2000 and agreed to return it with an additional 10% interest. This means she will pay a total of 110% of the principal amount</p> <p>Next, I calculate the total amount she needs to pay by multiplying the principal by 1.10: <math>\\$2000 * 1.10 = \\$2200</math></p> <p>Then, I determine how long it will take for her to pay this amount at a rate of \$165 per month. I set up the equation: <math>165 * t = 2200</math></p> <p>Solving for t gives: <math>t = 2200/165 = 13.333</math> months</p> <p>Since the payment is made every month, I calculate the total payment made after 13 months: <math>\\$165 * 13 = \\$2145</math></p> <p>Finally, I find the remaining balance by subtracting the total payment from the total amount owed: <math>\\$2200 - \\$2145 = \\$55</math>. &lt;/think&gt;</p> <p><b>**Solution:**</b></p> <p>1. <b>**Understand the Total Amount to be Paid:**</b>          Janeth borrowed \$2000 and promises to return it with an additional 10% of the amount. This means she will pay <b>**110%**</b> of the principal          Total Amount = <math>\\$2000 \times 1.10 = \\$2200</math></p> <p>2. <b>**Determine the Number of Months Needed to Pay \$2200 at \$165 per Month:**</b>          Let <math>t</math> be the number of months. The equation is: <math>165 \times t = 2200</math>          Solving for <math>t</math>  <math>t = \frac{2200}{165} \approx 13.\bar{3}</math> months</p> <p>3. <b>**Calculate the Total Payment After 13 Months:**</b>  <b>**Since payments are made every month: Total Payment = <math>\\$165 \times 13 = \\$2145</math></b></p> <p>4. <b>**Find the Remaining Balance:**</b>          Subtract the total payment from the total amount owed:          Remaining Balance = <math>\\$ 2200 - \\$2145 = \\$55</math></p> <p><b>**Final Answer:**</b> <span style="border: 1px solid black; padding: 2px;">55</span></p>

**Table 13** Examples of error labeled during manual annotation as “Resolution of complexity”

<p><b>Question 1</b></p> <p><b>gemma-2-9B</b> [Answered correctly in original question without math complexity]</p>	<p>Each person in a certain household consumes 0.2 kg of rice every meal. Supposing 5 members of the household eat rice every lunch and dinner, how many weeks will a exponential(log(42)) kg bag of rice last? [Correct Answer:3]</p> <p>1. * Rice consumption per meal per person: 0.2 kg * Number of people eating: 5 * Total rice consumption per meal: 0.2 kg/person * 5 people = 1 kg * Meals per day: 2 (lunch and dinner) * Total rice consumption per day: 1 kg/meal * 2 meals = 2 kg * Rice consumption per week: 2 kg/day * 7 days/week = 14 kg * Bag size: exponential(log(42)) kg * Weeks the rice will last: exponential(log(42)) kg/14 kg/week = ##### 1.9999999999999999</p>
<p><b>Question 2</b></p> <p><b>deepseek-llama-8B</b> [Answered correctly in original question without math complexity]</p>	<p>Darrell and Allen’s ages are in the ratio of 7:11. If their total age now is 162, calculate Allen’s age exponential(log(10)) years from now. [Correct Answer: 109]</p> <p>&lt;think&gt; First, I recognize that the ratio of Darrell’s age to Allen’s age is 7:11. This means that for every 7 years of Darrell’s age, Allen is 11 years</p> <p>Next, I’ll express their current ages in terms of a common variable. Let’s denote Darrell’s age as <math>7x</math> and Allen’s age as <math>11x</math>, where <math>x</math> is a common multiplier</p> <p>The problem states that the total of their current ages is 162 years. Therefore, I can set up the equation: <math>7x + 11x = 162</math></p> <p>Combining like terms gives: <math>18x = 162</math></p> <p>Solving for <math>x</math>: <math>x = 162/18 = 9</math></p> <p>Now that I have the value of <math>x</math>, I can find Allen’s current age: Allen’s age = <math>11x = 11 * 9 = 99</math> years</p> <p>The problem asks for Allen’s age in exponential(log(10)) years from now. First, I need to interpret what exponential(log(10)) means. The expression log(10) is the common logarithm, which is approximately equal to 1. Therefore, exponential(log(10)) is <math>e^1</math>, which is approximately 2.71828. Finally, to find Allen’s age in 2.71828 years, I add this value to his current age: Allen’s age in 2.71828 years = <math>99 + 2.71828 = 101.71828</math> years &lt;/think&gt;</p> <p>**Solution:**</p> <p>We are given that the ratio of Darrell’s age to Allen’s age is 7:11, and their combined ages now total 162 years. We need to determine Allen’s age in exponential(log(10)) years from now</p> <p>**Step 1: Express Their Current Ages in Terms of a Variable</p> <p>** Let’s denote: - Darrell’s current age as <math>7x</math> - Allen’s current age as <math>11x</math></p> <p>**Step 2: Set Up the Equation for Their Total Age**</p> $7x + 11x = 162$ <p>**Step 3: Solve for <math>x</math>**</p> $18x = 162$ $x = \frac{162}{18}$ $x = 9$ <p>**Step 4: Determine Allen’s Current Age**</p> <p>Allen’s age = <math>11x = 11 * 9 = 99</math> years</p> <p>**Step 5: Interpret exponential(log(10))**</p> <p>– log(10) is the common logarithm (base 10), which equals approximately 1. – Therefore, exponential(log(10)) = <math>e^1 \approx 2.71828</math></p> <p>**Step 6: Calculate Allen’s Age in 2.71828 Years**</p> <p>Allen’s age in 2.71828 years = <math>99 + 2.71828 \approx 101.71828</math> years</p> <p>**Final Answer:**</p> <div style="border: 1px solid black; padding: 2px; display: inline-block;">101.71828</div>

**Table 14** Example of error labeled during manual annotation as “Correctness of the final answer”

<b>Question</b>	Jerry is rolling a six-sided die. How much more likely is it (expressed as a percentage) that he rolls a number greater than $\log(\text{exponential}(3))$ than that he rolls two even numbers in a row? [Correct Answer: 25]
<b>gemma-9B</b> [Answered correctly in original question without math complexity]	1. <b>Calculate <math>\log(\text{exponential}(3))</math>:</b> Since the exponential function and the natural logarithm function are inverses of each other, $\log(\text{exponential}(3)) = 3$ . <b>Probability of rolling a number greater than 3:</b> There are no numbers greater than 3 on a standard six-sided die, so the probability is 0. <b>Probability of rolling two even numbers in a row:</b> * Probability of rolling an even number: $3/6 = 1/2$ * Probability of rolling another even number: $1/2$ * Probability of rolling two even numbers in a row: $(1/2) * (1/2) = 1/4$ <b>Difference in probabilities:</b> $0 - (1/4) = -1/4$ 2. ##### - 0.25

**Table 15** Example of error labeled during manual annotation as “Correctness of the answer pattern”

<b>Question</b>	Adrian’s age is three times the age of Harriet, and Harriet is half the age of Zack. Calculate the average age of the three in three years if Harriet is $\log(\text{exponential}(21))$ years old now. [Correct Answer: 45]
<b>gemma-9B</b> [Answered correctly in original question without math complexity]	1. Harriet’s age now = $\log(\text{exponential}(21)) = 21$ years (Since log and exponential are inverse functions) Adrian’s age now = $3 * \text{Harriet’s age} = 3 * 21 = 63$ years Zack’s age now = $2 * \text{Harriet’s age} = 2 * 21 = 42$ years Harriet’s age in three years = $21 + 3 = 24$ years Adrian’s age in three years = $63 + 3 = 66$ years Zack’s age in three years = $42 + 3 = 45$ years Average age in three years = $(\text{Harriet’s age} + \text{Adrian’s age} + \text{Zack’s age})/3 = (24 + 66 + 45)/3$ 2. 45

**Table 16** Examples of “Inverse” for llama-3.3-70B with complexity level equal to 2 and 4

<b>Question</b> [Level 2]	A robe takes $\text{exponential}(\log(\text{exponential}(\log(2))))$ bolts of blue fiber
<b>llama-3.3-70B</b> Generic prompt	<p>and half that much white fiber. How many bolts in total does it take? [Correct Answer: 3]</p> <p>1. First, calculate the innermost function: <math>\log(2)</math>.</p> <p>Since the base of the logarithm is not specified, we'll assume it's the common logarithm with base 10. <math>\log(2) \approx 0.301</math></p> <p>2. Next, calculate <math>\text{exponential}(\log(2))</math>: <math>\text{exponential}(0.301) = e^{0.301} \approx 1.349</math>.</p> <p>3. Now, calculate <math>\log(\text{exponential}(\log(2)))</math>: <math>\log(1.349) \approx 0.130</math>.</p> <p>4. Then, calculate <math>\text{exponential}(\log(\text{exponential}(\log(2))))</math>: <math>\text{exponential}(0.130) \approx e(0.130) \approx 1.139</math>.</p> <p>5. The robe takes 1.139 bolts of blue fiber and half that much white fiber, so the white fiber is <math>1.139/2 \approx 0.5695</math>.</p> <p>6. To find the total number of bolts, add the blue and white fiber: <math>1.139 + 0.5695 \approx 1.709</math>. #### 1.709</p>
<b>llama-3.3-70B</b> Notice prompt	<p>1. First, we need to simplify the expression <math>\text{exponential}(\log(\text{exponential}(\log(2))))</math> using the property that <math>\text{exponential}(\log(x)) \approx x</math>. So,</p> $\text{exponential}(\log(\text{exponential}(\log(2)))) = \text{exponential}(\log(\text{exponential}(\log(2))))$ $= \text{exponential}(\log(2))$ because $\text{exponential}(\log(2)) = 2$ . <p>Since <math>\text{exponential}(\log(2)) = 2</math>, then the total bolts of blue fiber is 2.</p> <p>The total bolts of white fiber is half of the total bolts of blue fiber, so it is <math>2/2 = 1</math>.</p> <p>The total bolts of fiber is the sum of the bolts of blue and white fiber, which is <math>2 + 1 = 3</math>. #### 3</p>
<b>Question</b> [Level 4]	A robe takes $\text{exponential}(\log(\text{exponential}(\log(\text{exponential}(\log(\text{exponential}(\log(2))))))))$
<b>llama-3.3-70B</b> Generic prompt	<p>bolts of blue fiber and half that much white fiber. How many bolts in total does it take? [Correct Answer: 3]</p> <p>1. Start with the innermost function: <math>\text{exponential}(\log(2))</math>. Since exponential and logarithm are inverse operations, <math>\text{exponential}(\log(2)) = 2</math>. 2. Now we have <math>\text{exponential}(\log(\text{exponential}(\log(2)))) = \text{exponential}(\log(2)) = 2</math>. 3. Again, <math>\text{exponential}(\log(\text{exponential}(\log(\text{exponential}(\log(2)))))) = \text{exponential}(\log(2)) = 2</math>.</p> <p>4. Continuing this pattern, <math>\text{exponential}(\log(\text{exponential}(\log(\text{exponential}(\log(\text{exponential}(\log(\text{exponential}(\log(2)))))))))) = 2</math></p> <p>5. The robe takes 2 bolts of blue fiber. 6. It takes half that much white fiber, so it takes <math>2/2 = 1</math> bolt of white fiber.</p> <p>7. Total bolts = bolts of blue fiber + bolts of white fiber = <math>2 + 1 = 3</math>. #### 3</p>

## Appendix D: Comparison of Errors Across Models

We measured the number of errors that are shared by pairs of top-performing models on the GSM8K questions, for the various modification types and prompts (see Table 17).

**Table 17** Number of errors shared by pairs of models on the GSM-Identity questions with Complex 5-CoT prompt

		qwen2.5-7B	gemma-2-9B	llama-3.3-70B	qwen2.5-math-72B
When the pair is composed of the same model, the number indicated the number of errors it made on the task. The count is split by type of added complexity: BODMAS, Inverse, Trigonometry	qwen2.5-7B	BODMAS 183	103	52	44
		Inverse 171	111	47	46
		Trigonometry 184	105	60	53
	gemma-2-9B	BODMAS 103	266	57	43
		Inverse 111	359	53	50
		Trigonometry 105	303	60	50
	llama-3.3-70B	BODMAS 52	57	80	32
		Inverse 47	53	70	28
		Trigonometry 60	60	81	38
qwen2.5-math-72B	BODMAS 44	43	32	67	
	Inverse 46	50	28	74	
	Trigonometry 53	50	38	87	

**Table 18** Number of errors shared by gemma-2-9B-it and qwen2.5-7B for GSM-Identity questions

		qwen2.5-7B				
		Generic	Notice	5-CoT	Complex 5-CoT	
gemma-2-9B	Generic	BODMAS	95	95	93	107
		Inverse	261	174	203	131
		Trigonometry	120	115	124	124
	Notice	BODMAS	99	98	100	114
		Inverse	218	154	163	120
		Trigonometry	96	96	110	112
	5-CoT	BODMAS	93	91	99	107
		Inverse	229	153	193	127
		Trigonometry	112	103	113	109
	Complex 5-CoT	BODMAS	98	97	101	103
		Inverse	157	111	137	119
		Trigonometry	98	102	111	105

BODMAS (Top) Inverse (Middle) Trigonometry (Bottom)

**Table 19** Number of errors shared by qwen2.5-math-72B and llama-3.3-70B for GSM-Identity questions

			qwen2.5-math-72B			
			Generic	Notice	5-CoT	Complex 5-CoT
llama-3.3-70B	Generic	BODMAS	34	33	36	34
		Inverse	31	29	32	29
		Trigonometry	38	38	41	37
	Notice	BODMAS	33	30	33	31
		Inverse	30	29	31	28
		Trigonometry	37	37	37	35
	5-CoT	BODMAS	31	32	30	32
		Inverse	32	30	31	27
		Trigonometry	34	36	35	35
	Complex 5-CoT	BODMAS	30	30	32	32
		Inverse	27	28	27	28
		Trigonometry	35	38	38	38

BODMAS (Top) Inverse(Middle) Trigonometry (Bottom)

## Appendix E: Detailed Results on Additional Datasets

Table 20 show the accuracy values for the 192 configurations tested on the additional datasets (see Sect. 7).

**Table 20** Accuracy of the models with different prompting for original questions and the respective GSM-Identity questions on the three-*platinum* datasets

Dataset	Model	Prompt	Original	BODMAS	Inverse	Trigonometry	Avg
multiarith-platinum	gemma-2-9B	Generic	94.3	89.1	37.9	62.1	63.0
		Notice	89.7	86.2	57.5	86.8	76.8
		5-CoT	92.5	85.1	67.8	78.7	77.2
		Complex 5-CoT	94.3	90.8	86.8	83.3	87.0
	qwen2.5-7B	Generic	97.7	96.6	74.7	94.8	88.7
		Notice	97.7	92.5	86.2	94.3	91.0
		5-CoT	97.1	94.8	86.2	94.8	91.9
		Complex 5-CoT	97.7	94.8	90.2	92.0	92.3
	llama-3.3-70B	Generic	97.1	96.0	87.9	93.7	92.5
		Notice	97.7	96.6	96.0	96.6	96.4
		5-CoT	97.1	96.6	95.4	96.6	96.2
		Complex 5-CoT	97.1	97.1	97.1	97.1	97.1
	qwen2.5-math-72B	Generic	97.7	97.7	96.0	97.7	97.1
		Notice	97.7	97.7	97.1	97.7	97.5
		5-CoT	97.7	97.7	97.7	97.7	97.7
		Complex 5-CoT	97.7	97.1	97.7	97.7	97.5
singleq-platinum	gemma-2-9B	Generic	97.0	94.0	27.0	67.0	62.7
		Notice	96.0	94.0	48.0	92.0	78.0
		5-CoT	97.0	96.0	69.0	85.0	83.3
		Complex 5-CoT	100.0	100.0	77.0	95.0	90.7
	qwen2.5-7B	Generic	95.0	100.0	84.0	99.0	94.3
		Notice	92.0	100.0	89.0	100.0	96.3
		5-CoT	100.0	99.0	92.0	100.0	97.0
		Complex 5-CoT	100.0	100.0	93.0	100.0	97.7
	llama-3.3-70B	Generic	97.0	100.0	84.0	98.0	94.0
		Notice	97.0	100.0	99.0	100.0	99.7
		5-CoT	100.0	100.0	99.0	100.0	99.7
		Complex 5-CoT	100.0	100.0	100.0	100.0	100.0
	qwen2.5-math-72B	Generic	100.0	100.0	100.0	100.0	100.0
		Notice	100.0	100.0	99.0	100.0	99.7
		5-CoT	100.0	100.0	97.0	100.0	99.0
		Complex 5-CoT	100.0	100.0	100.0	100.0	100.0

**Table 20** (continued)

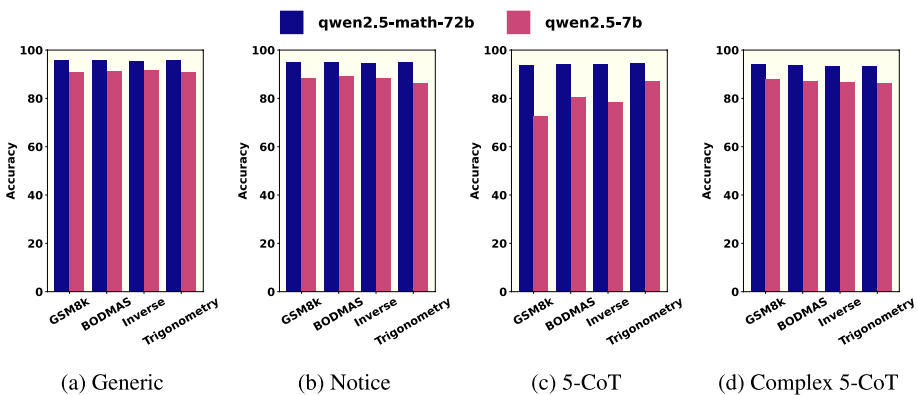
Dataset	Model	Prompt	Original	BODMAS	Inverse	Trigonometry	Avg
svamp-platinum	gemma-2-9B	Generic	88.4	84.7	36.2	64.9	61.9
		Notice	87.7	82.5	53.4	85.1	73.7
		5-CoT	88.8	73.5	66.4	78.7	72.9
		Complex 5-CoT	88.4	86.6	80.2	85.4	84.1
	qwen2.5-7B	Generic	91.0	92.9	77.6	89.6	86.7
		Notice	91.4	92.9	82.8	91.4	89.0
		5-CoT	95.5	92.5	83.6	92.2	89.4
		Complex 5-CoT	94.8	90.7	93.3	90.7	91.6
	llama-3.3-70B	Generic	96.3	96.3	91.4	93.7	93.8
		Notice	97.4	97.0	94.4	97.0	96.1
		5-CoT	97.4	96.6	94.0	97.0	95.9
		Complex 5-CoT	96.6	96.6	96.6	97.0	96.7
	qwen2.5-math-72B	Generic	96.3	97.4	95.9	96.3	96.5
		Notice	97.4	98.1	96.6	97.0	97.2
		5-CoT	97.0	96.3	97.0	94.8	96.0
		Complex 5-CoT	97.0	96.3	95.5	95.5	95.8

The average is calculated only on GSM-Identity questions

## Appendix F: Inference Comparison

See Tables 21, 22 and Fig. 8.

During our experiments, we switched to vLLM due to its much faster inference, as the Hugging Face implementation required days to complete the experiments on a single data-



**Fig. 8** Difference of accuracy of the two top-performing models in their one class size, i.e., qwen2.5-math-72B (70B) vs qwen2.5-7B (7B). GSM8K and GSM-Identity datasets, different prompts: **a** Generic, **b** Notice, **c** 5-CoT, **d** Complex 5-CoT

**Table 21** Average loss in accuracy compared to original GSM8K for llama-3.3-70B in different complexity level across different prompts

Level	Generic	Notice	5-CoT	Complex 5-CoT
1	4.2	1.3	1.6	1.0
2	6.6	4.0	4.5	3.1
3	7.2	4.2	4.2	3.4
4	6.7	4.4	5.6	4.4
5	8.2	6.0	6.0	5.1

**Table 22** Comparison in model accuracy using vLLM and Hugging Face

Model	Prompt Type	Accuracy(vLLM)	Accuracy(HF)	Absolute Accuracy % difference
deepseek-llama-8B	Generic	0.666	0.748	8.2
	Notice	0.697	0.753	5.60
	5-CoT	0.729	0.771	4.2
	Complex 5-CoT	0.701	0.739	3.8
mistral-nemo	Generic	0.706	0.801	9.5
	Notice	0.710	0.845	13.5
	5-CoT	0.809	0.845	3.6
	Complex 5-CoT	0.815	0.848	3.3
gemma-2-9B	Generic	0.858	0.867	0.9
	Notice	0.848	0.873	2.5
	5-CoT	0.850	0.873	2.3
	Complex 5-CoT	0.864	0.867	0.3

set, while vLLM required few hours. However, we observed a difference in accuracy in our results, even though the trends remained the same across different prompts. Table 22 shows the difference in the count of correct answers for the original GSM8K dataset when using vLLM compared to Hugging Face.

**Acknowledgements** This work was partially supported by FAIR (PE00000013) project under the NextGenerationEU programme, partially by the PNRR project ITSERR (CUP B53C22001770006) and partially by the Project PRIN 2022EPTP19 (WEMB – “Word EMBeddings: From Cognitive Linguistics to Language Engineering, and Back”), funded by the Italian Ministry of University and Research (MUR).

**Author Contributions** K.N.: Methodology, Software, Validation, Investigation, Data Curation, Writing G.P.: Conceptualization, Methodology, Software, Investigation, Writing, Supervision A.E.: Conceptualization, Methodology, Validation, Investigation, Writing, Supervision, Project administration

**Funding** Open access funding provided by Consiglio Nazionale Delle Ricerche (CNR) within the CRUI-CARE Agreement. This work was partially supported by FAIR (PE00000013) project under the NextGenerationEU programme, partially by the PNRR project ITSERR (CUP B53C22001770006) and partially by the Project PRIN 2022EPTP19 (WEMB—“Word EMBeddings: From Cognitive Linguistics to Language Engineering, and Back”), funded by the Italian Ministry of University and Research (MUR).

**Data Availability** All the external datasets used in the paper are publicly available. The GSM-Identity data is generated by the code we made available.

**Code Availability** The method and the code to replicate the experiments is available under an open-source license at [https://github.com/kajalnegi/complex\\_math\\_evaluation.git](https://github.com/kajalnegi/complex_math_evaluation.git).

## Declarations

**Conflict of interest** Authors have no conflict of interest or conflict of interest with respect to the content of this article.

**Ethical Approval** Not applicable.

**Consent to Participate** Not applicable.

**Consent for Publication** Not applicable.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Baevski, A., Zhou, H., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: a framework for self-supervised learning of speech representations. In *Proceedings of the 34th international conference on neural information processing systems*. Curran Associates.
- Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency* (pp. 610–623). Association for Computing Machinery.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., & Amodei, D. (2020). Language models are few-shot learners. In *Advances in neural information processing systems* (Vol. 33, pp. 1877–1901). Curran Associates.
- Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., & Schulman, J. (2021). *Training verifiers to solve math word problems*. [arXiv:2110.14168](https://arxiv.org/abs/2110.14168)
- DeepSeek-AI (2025). *Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning*. [arXiv.org/abs/2501.12948](https://arxiv.org/abs/2501.12948)
- Dong, Q., Li, L., Dai, D., Zheng, C., Ma, J., Li, R., & Sui, Z. (2024). A survey on in-context learning. In Y. Al-Onaizan, M. Bansal, & Y.-N. Chen (Eds.), *Proceedings of EMNLP-2024* (pp. 1107–1128). Miami, Florida, USA.
- Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., & Ma, Z. (2024). *The llama 3 herd of models*. [arXiv.org/abs/2407.21783](https://arxiv.org/abs/2407.21783)
- Gulati, A., Miranda, B., Chen, E., Xia, E., Fronsdal, K., de Moraes Dumont, B., & Koyejo, S. (2024). Putnam-AXIOM: A functional and static benchmark for measuring higher level mathematical reasoning. In *The 4th workshop on mathematical reasoning and AI at NeurIPS'24*. <https://openreview.net/forum?id=YXnwlZe0yf>
- Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., & Steinhardt, J. (2021). Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 2)*. <https://openreview.net/forum?id=7Bywt2mQsCe>
- Huang, K., Guo, J., Li, Z., Ji, X., Ge, J., Li, W., & Wang, M. (2025). *MATH-Perturb: Benchmarking llms' math reasoning abilities against hard perturbations*. Retrieved March 27, 2025, from [arXiv:2502.06453](https://arxiv.org/abs/2502.06453) [cs]
- Hui, B., Yang, J., Cui, Z., Yang, J., Liu, D., Zhang, L., & Lin, J. (2024). *Qwen2.5-coder technical report*. [arXiv.org/abs/2409.12186](https://arxiv.org/abs/2409.12186)

- Hupkes, D., Giulianelli, M., Dankers, V., Artetxe, M., Elazar, Y., Pimentel, T., & Jin, Z. (2023). A taxonomy and review of generalization research in NLP. *Nature Machine Intelligence*, 5(10), 1161–1174.
- Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C.H., & Stoica, I. (2023). Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles* (pp. 611–626). Association for Computing Machinery.
- Li, Q., Cui, L., Zhao, X., Kong, L., & Bi, W. (2024). *Gsm-plus: A comprehensive benchmark for evaluating the robustness of LLMs as mathematical problem solvers*. [arXiv.org/abs/2402.19255](https://arxiv.org/abs/2402.19255)
- Li, Y.A., Han, C., Raghavan, V., Mischler, G., & Mesgarani, N. (2023). Styletts 2: Towards human-level text-to-speech through style diffusion and adversarial training with large speech language models. In *Advances in neural information processing systems* (Vol. 36, pp. 19594–19621). Curran Associates.
- Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., & Van Gool, L. (2022). Repaint: Inpainting using denoising diffusion probabilistic models. In *2022 IEEE/CVF conference on computer vision and pattern recognition (CVPR)* (pp. 11451–11461).
- McHugh, M. L. (2012). Interrater reliability: The kappa statistic. *Biochemia medica*, 22(3), 276–282.
- Mirzadeh, S. I., Alizadeh, K., Shahrokhii, H., Tuzel, O., Bengio, S., & Farajtabar, M. (2025). GSM-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The 13th international conference on learning representations*. <https://openreview.net/forum?id=AjXkrZlVjB>
- Mistral AI Team. (2023). *Mistral 7b*. [arXiv.org/abs/2310.06825](https://arxiv.org/abs/2310.06825)
- Mitchell, M. (2021). Why AI is harder than we think. In *Proceedings of the genetic and evolutionary computation conference* (p. 3). Association for Computing Machinery. <https://doi.org/10.1145/3449639.3465421>
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. <https://api.semanticscholar.org/CorpusID:160025533>
- Ray Choudhury, S., Rogers, A., & Augenstein, I. (2022). Machine reading, fast and slow: When do models “understand” language? In *Proceedings of coling-2022* (pp. 78–93). International Committee on Computational Linguistics.
- Shah, V., Yu, D., Lyu, K., Park, S., Ke, N. R., Mozer, M. C., & Goyal, A. (2024). AI-assisted generation of difficult math questions. In *The 4th workshop on mathematical reasoning and AI at NEURIPS'24*. <https://openreview.net/forum?id=6pUdfsJmd1>
- Shi, F., Chen, X., Misra, K., Scales, N., Dohan, D., Chi, E., & Zhou, D. (2023). Large language models can be easily distracted by irrelevant context. *Proceedings of the 40th international conference on machine learning*. [www.jmlr.org](http://www.jmlr.org).
- Srivastava, A., Rastogi, A., Rao, A., Shoeb, A.A.M., Abid, A., Fisch, A., & Wu, Z. (2023). Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*. <https://openreview.net/forum?id=uyTL5Bvosj> (Featured Certification)
- Team, G., Mesnard, T., Hardin, C., Dadashi, R., Bhupatiraju, S., Pathak, S., & Kenealy, K. (2024). *Gemma: Open models based on gemini research and technology*. [arXiv.org/abs/2403.08295](https://arxiv.org/abs/2403.08295)
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., & Lample, G. (2023). *Llama: Open and efficient foundation language models*. [arXiv.org/abs/2302.13971](https://arxiv.org/abs/2302.13971)
- Vendrow, J., Vendrow, E., Beery, S., & Madry, A. (2025). *Do large language model benchmarks test reliability?*. [arXiv.org/abs/2502.03461](https://arxiv.org/abs/2502.03461)
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F. & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th international conference on neural information processing systems*. Curran Associates Inc.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., & Rush, A. (2020). Transformers: State-of-the-art natural language processing. In Q. Liu & D. Schlangen (Eds.), *Proceedings of EMNLP-2020* (pp. 38–45).
- Xu, R., Wang, Z., Fan, R.-Z., & Liu, P. (2024). *Benchmarking benchmark leakage in large language models*. [arXiv.org/abs/2404.18824](https://arxiv.org/abs/2404.18824)
- Yang, A., Zhang, B., Hui, B., Gao, B., Yu, B., Li, C., & Zhang, Z. (2024). *Qwen2.5-math technical report: Toward mathematical expert model via self-improvement*. [arxiv.org/abs/2409.12122](https://arxiv.org/abs/2409.12122)
- Zhou, Y., Liu, H., Chen, Z., Tian, Y., & Chen, B. (2025). *GSM-infinite: How do your LLMs behave over infinitely increasing context length and reasoning complexity?*. [arXiv.org/abs/2502.05252](https://arxiv.org/abs/2502.05252)