

Exposing Numeracy Gaps: A Benchmark to Evaluate Fundamental Numerical Abilities in Large Language Models

Haoyang LI¹, Xuejia CHEN¹, Zhanchao XU¹, Darian LI¹, Nicole HU³, Fei TENG², Yiming LI^{2*}, Luyu QIU², Chen Jason ZHANG¹, Qing LI¹, Lei CHEN²

¹Hong Kong Polytechnic University,

²Hong Kong University of Science and Technology,

³The Chinese University of Hong Kong

haoyang-comp.li@polyu.edu.hk, yliix@connect.ust.hk

Abstract

Large Language Models (LLMs) have demonstrated impressive capabilities in natural language processing tasks, such as text generation and semantic understanding. However, their performance on numerical reasoning tasks, such as basic arithmetic, numerical retrieval, and magnitude comparison, remains surprisingly poor. This gap arises from their reliance on surface-level statistical patterns rather than understanding numbers as continuous magnitudes. Existing benchmarks primarily focus on either linguistic competence or structured mathematical problem-solving, neglecting fundamental numerical reasoning required in real-world scenarios. To bridge this gap, we propose NumericBench, a comprehensive benchmark to evaluate six fundamental numerical capabilities: number recognition, arithmetic operations, contextual retrieval, comparison, summary, and logic reasoning. NumericBench includes datasets ranging from synthetic number lists to the crawled real-world data, addressing challenges like long contexts, noise, and multi-step reasoning. Extensive experiments on state-of-the-art LLMs, including GPT-4 and DeepSeek, reveal persistent weaknesses in numerical reasoning, highlighting the urgent need to improve numerically-aware language modeling. The benchmark is released in: <https://github.com/TreeAI-Lab/NumericBench>.

1 Introduction

Large language models (LLMs) (Zhao et al., 2024) have demonstrated remarkable capabilities in text generation, semantic understanding, and task adaptation across diverse domains (Ling et al., 2024). Their success is largely attributed to pretraining on vast text corpora using next-token prediction objectives (He and Su, 2024), which enables generalization to tasks requiring linguistic creativity, commonsense reasoning, and domain-specific knowledge (Ye, 2024). However, while LLMs perform

*Corresponding Author

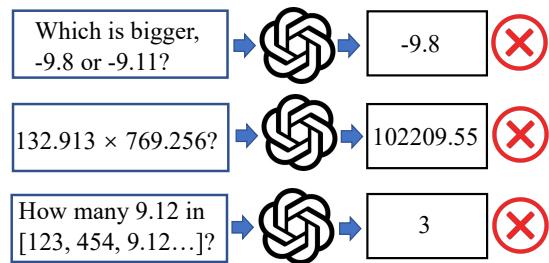


Figure 1: Numerical tasks answered incorrectly by GPT-4o. Details are in Figure 5, Figure 6, and Figure 7.

well in text understanding tasks and Olympic mathematics questions (Team et al., 2024), they surprisingly struggle with simple numerical tasks such as basic multiplication, comparison, and retrieval. As shown in Figure 1, effective GPT-4o fails at simple numerical tasks, even with number comparisons.

Unlike tasks that rely primarily on semantic coherence and linguistic structures, numerical reasoning requires a deeper understanding of numbers as continuous magnitudes rather than discrete tokens. Current LLMs tend to prioritize surface-level statistical patterns, such as lexical co-occurrences and syntactic regularities, which limits their ability to process numerical operations (Ahn et al., 2024; Feng et al., 2024; Zhou et al.). As a result, LLMs frequently struggle with tasks involving numeric retrieval, arithmetic operations, and magnitude comparisons. These shortcomings highlight an urgent need to systematically evaluate and improve the numerical reasoning capabilities of LLMs.

Current evaluation frameworks for LLMs prioritize either linguistic competence or formal mathematical problem-solving. For instance, semantic-oriented benchmarks (Vulić et al., 2020), such as GLUE (Wang, 2018), SuperGLUE (Wang et al., 2019), and SimpleQA (Wei et al., 2024), primarily assess linguistic competence and semantic understanding, while math-oriented benchmarks, such as MathQA (Amini et al., 2019), GSM8K (Cobbe et al., 2021), and MathBench (Liu et al., 2024b), focus on structured algebraic or geometric tasks.

However, these approaches neglect the basic demands of real-world numerical reasoning applications and fundamental numerical abilities, where numbers are often embedded in unstructured and noisy context data. For example, analyzing fluctuating stock prices or weather requires basic numeric retrieval, comparison, and summary abilities.

To address the limitations of existing benchmarks, we propose a comprehensive benchmark NumericBench, which consists of six general datasets, i.e., arithmetic numbers, mixed-number-strings, number lists, stock, weather, and numerical sequences with patterns. Unlike prior benchmarks, NumericBench systematically evaluate the six fundamental numerical abilities of LLMs: **(1) Number Recognition:** It evaluates the ability of LLMs to identify numbers within dense strings. **(2) Arithmetic Operation:** It tests basic arithmetic operations, including addition, subtraction, multiplication, and division. **(3) Contextual Retrieval:** It evaluates LLMs to retrieve specific numerical values from a given context or a number list. **(4) Comparison:** It determine relationships between values, such as comparing price differences. **(5) Summary:** It target to summarize trends (e.g., determining the number of consecutive days a stock price increases) and data aggregation. **(6) Logic Reasoning:** It tests whether LLMs can understand numerical patterns and predict the next value, which is widely used in time-series forecasting, such as weather and traffic prediction.

By integrating six datasets, ranging from synthetic numerical lists to crawled real-world data, NumericBench evaluates six fundamental numerical abilities of LLMs across various scenarios, such as noisy contexts. Our experimental analysis over various effective series of LLMs, including GPT-4 (Achiam et al., 2023), DeepSeek (Liu et al., 2024a), and Llama (Touvron et al., 2023), reveals persistent weaknesses in handling these fundamental numerical tasks. We further analyze five potential reasons behind these numerical reasoning gaps, including tokenizer, training corpora, training paradigms, positional embeddings, and architectural constraints. These findings offer actionable insights to guide future numerical ability improvements for LLMs. Given that numerical reasoning is critical for real-world problem-solving, it represents a cornerstone for the development of Artificial General Intelligence (AGI). This emphasizes the urgent need to advance numerically-aware language modeling. The contributes of this paper are

summarized as follows:

- We propose a comprehensive NumericBench, which integrates diverse datasets and reflects real-world challenges, such as handling noisy or domain-specific data (e.g., stock and weather). NumericBench evaluates six fundamental numerical reasoning abilities, including number recognition, arithmetic operations, contextual retrieval, comparison, summary, and logic reasoning.
- Extensive experiments on effective LLMs (e.g., GPT-4, DeepSeek, Llama) reveal persistent weaknesses in numerical reasoning tasks, including basic arithmetic, comparison, and logic reasoning. This highlights the need for more numerically-aware modeling approaches.
- We identify key challenges behind the numerical reasoning gaps in LLMs, such as tokenization practices, training paradigms, positional embeddings, and architectural constraints. These insights provide actionable guidance for future model improvements.

2 Preliminary and Related Works

In this section, we first introduce large language models and then present existing benchmarks.

2.1 Large Language Models

Large language models (LLMs), such as GPT-4 (Achiam et al., 2023), DeepSeek (Liu et al., 2024a), PaLM (Anil et al., 2023), and Llama (Touvron et al., 2023), have revolutionized natural language processing (NLP) through their ability to generate coherent text (Cho et al., 2019), answer questions (Chen et al., 2024), and adapt to diverse tasks (Wang et al., 2025; Jiang et al., 2024). Their success stems from pretraining on vast text corpora using next-token prediction objectives, which enable generalization on tasks requiring semantic understanding, commonsense reasoning, and linguistic creativity. However, this training paradigm encourages LLMs to prioritize surface-level statistical patterns (e.g., lexical co-occurrences, syntactic regularities) rather than numerically grounded reasoning (Bachmann and Nagarajan, 2024). Consequently, LLMs treat numbers as discrete tokens rather than continuous magnitudes, inherently limiting their ability to understand exact numerical semantics. This leads to errors in numeric retrieval, arithmetic operations, and magnitude comparisons (Qiu et al., 2024).

2.2 Benchmarks on Large Language Models

Existing benchmarks (Li et al., 2024a; Chang et al., 2024; Zhao et al., 2023) for evaluating LLMs primarily fall into two categories, i.e., *semantic-oriented* and *math-oriented* benchmarks. Specifically, *semantic-oriented* benchmarks, such as GLUE (Wang, 2018), SuperGLUE (Wang et al., 2019), SimpleQA (Wei et al., 2024), and LongBench (Bai et al., 2023), focus on semantic understanding and linguistic competence, testing skills like textual entailment, commonsense reasoning, and domain-specific knowledge (e.g., science and law). While effective for assessing linguistic proficiency, these benchmarks largely overlook numerical reasoning. On the other hand, *math-oriented* benchmarks (Gao et al., 2025; Li et al., 2024b; Cobbe et al., 2021), such as MathQA (Amini et al., 2019), GSM8K (Cobbe et al., 2021), and MathBench (Liu et al., 2024b), target mathematical problem-solving (e.g., algebra, calculus) or extractive question-answering with numerical answers. However, these datasets emphasize well-formed mathematical problems in controlled and clean settings. Consequently, *math-oriented* benchmarks poorly evaluate numerical retrieval and reasoning in real-world conditions, where noise, and contextual complexity (e.g., multi-step financial workflows or long stock sequences) are common.

Considering that numeric retrieval and reasoning are critical for real-world applications (Yang et al., 2025), such as finance (Islam et al., 2023) and weather forecasting (Zhang et al., 2024), we propose **NumericBench** to systematically evaluate the fundamental numerical abilities of LLMs on intensive tasks, such as precise value retrieval, dynamic comparisons, and arithmetic-logic reasoning.

3 NumericBench

In this section, we present our created NumericBench, which is specifically designed to evaluate fundamental numerical capabilities of LLMs. NumericBench consists of diverse datasets and tasks, enabling a systematic and comprehensive evaluation. We discuss the datasets included in NumericBench, the key abilities it evaluates, and the methodology for benchmark generation.

3.1 Numeric Dataset Collection

NumericBench offers a diverse collection of numerical datasets and questions designed to reflect real-world scenarios. This variety ensures that LLMs

are thoroughly tested on their fundamental abilities on numerical data.

Number List Dataset. The synthetic number list dataset consists of randomly generated numerical values, including both integers and floating-point numbers. Presented as ordered or unordered lists. Numbers in lists are one of the simplest and most fundamental data representations encountered in real-world scenarios. Despite their simplicity, retrieving, indexing, comparison, and summary on numbers can verify the fundamental numerical ability of LLMs. This dataset serves as a fundamental dataset of how well LLMs understand numerical values as discrete entities.

Stock Dataset. The time-series stock dataset is crawled from Eastmoney website (Eastmoney Website, 2025), which has eighteen attributes, such as stock close prices, open prices, trading volumes, and price-earnings ratios, over time. Stock data reflects dynamic, real-world numerical reasoning challenges that involve trend analysis, comparison, and decision-making under uncertainty, representing real-world financial workflows.

Weather Dataset. The weather dataset is crawled from Open-Meteo python API (Open-Meteo), which includes data related to weather metrics, such as temperature, precipitation, humidity, and wind speed. The data is presented across various longitude and latitude.

Numeric Sequence Dataset. The synthetic numeric sequence dataset comprises sequences of numbers generated by arithmetic or geometric sequence with various patterns, such as Fibonacci Sequence. Tasks require identifying patterns, predicting the next number, or reasoning about relationships between numbers. Numerical sequences test the logic reasoning capabilities of LLMs, requiring pattern recognition and multi-step reasoning. We introduce structured challenges that are common in mathematics and algorithmic reasoning.

Arithmetic Operation Dataset. The dataset comprises 12,000 pairs of simple numbers, each undergoing addition, subtraction, multiplication, and division operations. Each pair of numbers, a and b , consists of k -digit integers with three decimal places, where $k \in \{1, 2, \dots, 6\}$. For each value of k , there are 2,000 pairs, evenly distributed across the four basic operations (i.e., $+, -, *, /$), with 500 pairs per operation. This dataset is to evaluate the fundamental mathematical operation capabilities of LLMs, simulating the majority of mathematical calculation requirements in real-world scenarios.

Table 1: NumericBench statistics. R: contextual retrieval, C: comparison, S: summary, L: logic reasoning. The token count is calculated based on tiktoken, which is the tokenizer used by Llama3 (Grattafiori et al., 2024). The sentences used for token calculation include both the context and the question.

Data	Format	Questions	# Instance	Avg Token
Number List	[69, -1, 6.1, ..., 5.7]	R: What is the index of the first occurrence of the number -3095 in the list?	500	3704.23
		C: Which index holds the smallest number in the list between the indices 20 and 80?	500	3685.57
		S: What is the average of the index of top 30 largest numbers in the list?	500	3654.78
Stock	{date: 2024-06-19, close_price: 9.79, open_price: 9.4, ..., PE_ratio: 4.5416}	R: On which date did the close price of stock firstly reach 61.76 yuan?	500	27585.35
		C: Among the top-45 trading value days, which date did the stock have the lowest close price?	500	27595.40
		S: How many days had the close price higher than the open price from 2024-07-31 to 2024-12-13?	500	27561.29
Weather	{date: 2024-07-21, pressure_msl: 999.96, dew_point_2m: 26.25, ..., cloud_cover: 61.5}	R: On which date did the dew point temperature at two meters lastly drop below 9.15°C?	500	27359.26
		C: On which date did the MSL pressure reach its highest value when the cloud cover was below 9%?	500	27368.19
		S: What was the average temperature at two meters when the relative humidity exceeded 78.56%?	500	27331.21
Sequence	[0.34, 3, 6, ..., 111]	L: What is the next number in the sequence?	500	677.57
Arithmetic Operation	$a : 6.755, b : -1.225$	Q_{oper} : What is the result of $a + b$? $Q_{context}$: What is the result of a plus b ?	12000	112.00
Mixed-number-string Sequence	$effV2 \dots x98o7Lo$	How many numbers are there in the string? Note that a sequence like 'a243b' counts as a single number.	2000	196.53

Mixed-number-string Sequence Dataset. The dataset consists of alphanumeric strings of varying lengths {50, 100, 150, 200}, each containing a randomized mix of letters and digits. For each string length, 500 samples are generated, resulting in a total of 2,000 samples. Each sample includes a query asking for the count of contiguous numeric sequences within the string, where sequences like "a243b" count as a single number. This dataset is designed to assess the ability of LLMs to identify and count numeric sequences.

3.2 Fundamental Numerical Ability

NumericBench is designed to comprehensively evaluate six fundamental numerical reasoning abilities of LLMs, which are essential for solving real-world numeric-related tasks.

Contextual Retrieval Ability. Contextual retrieval ability evaluates how well LLMs can locate, extract, and identify specific numerical values or their positions within structured or unstructured data. This includes tasks like finding a specific number

in a list, retrieving values , and indexing numbers based on their order. For example, as shown in Table 1, it evaluates LLMs on tasks such as retrieving stock prices and identifying key values within numerical lists or domain-specific data (e.g., stock market and weather-related information). This ability is fundamental to numerical reasoning because it forms the foundation for higher-order tasks, such as comparison, aggregation, and logic reasoning.

Comparison Ability. Comparison ability evaluates how well LLMs can compare numerical values to determine relationships such as greater than, less than, or equal to, and identify trends or differences in datasets. Comparison is vital for logic reasoning and decision-making, as many real-world tasks depend on accurate numerical evaluation. For instance, as shown in Table 1, comparing prices is essential in stock for assessing performance, while weather forecasting requires analysis of temperature or precipitation trends over time.

Summary Ability. Summary ability assesses the LLM’s capacity to aggregate numerical data into

concise insights, such as calculating totals, averages, or other statistical metrics. Summarization is critical for condensing large datasets into actionable information, enabling decision-making based on aggregated insights rather than raw data. This ability is indispensable in domains like electricity usage analysis, where summarizing hourly or daily consumption helps forecast bills, in business reporting for aggregating sales and revenue data to evaluate performance, and in healthcare analytics to monitor trends in patient metrics over time.

Logic Reasoning Ability. Logic Reasoning Ability measures the LLM’s ability to perform multi-step operations involving numerical data, such as recognizing patterns, inferring rules, and applying arithmetic or geometric reasoning to solve complex problems. Logic reasoning extends beyond simple numerical tasks and reflects the LLM’s capacity for deeper, structured thinking. This ability is crucial for algorithm design, where solving problems involving numeric sequences or patterns is essential. It is also required in scientific research for identifying relationships and correlations in data.

Arithmetic Operation Ability. It reflects the LLM’s capacity to perform mathematical calculations accurately. Such ability is essential for tasks involving numerical computations, such as automated machine learning through LLMs.

Number Recognition Ability. This measures the LLM’s proficiency in identifying and interpreting numerical information within a given context. It represents a fundamental requirement for handling numeric-based tasks effectively.

3.3 NumericBench Generation

We use the number list, stock, and weather datasets to evaluate the contextual retrieval, comparison, and summary abilities of LLMs. Specifically, for each ability and each dataset, we prepare a set of questions designed to assess the corresponding target ability. As shown in Table 8, Table 9, and Table 10 in Appendix, there are nine question sets in total, covering three abilities across three datasets. When evaluating a specific ability (e.g., contextual retrieval) on a specific dataset (e.g., stock data), we randomly select one question from the corresponding question set for each data instance (e.g., a stock instance) and manually label the answer. This approach enables us to generate question-answer pairs for each ability on the number list, stock, and weather datasets.

Moreover, we generate a logic reasoning dataset

with 500 long sequences using general term formulas. The target value of each sequence is removed to form an inference task, as detailed in Table 1 of our paper. For arithmetic operations and number counting in the strings dataset, the question format is straightforward, as illustrated in Table 1. These questions evaluate the basic arithmetic operation and number recognition abilities of LLMs.

4 Experiments

4.1 Experiment Setting

Benchmarks and Evaluated Protocols. The statistic of NumericBench is provided in Table 1. Also, we set the exact answer for mixed-number-string dataset, set the computed answer to two decimal places for arithmetic datasets, and set the answer of each question as a single choice (e.g., A, B, or C) for other datasets to reliably evaluate LLMs (Bai et al., 2024). The evaluation metric is accuracy.

Evaluated Models. To comprehensively evaluate the retrieval and reasoning abilities of state-of-the-art and widely-used LLMs on numeric data, we benchmark over 10 popular LLMs with our constructed NumericBench, as follows.

- **The Llama Series (Grattafiori et al., 2024).** include Llama-3.1-8B-Instruct, Llama-3.1-70B-Instruct, Llama-3.1-405B-Instruct, Llama-3.3-70B-Instruct and Llama-3.1-Nemotron-70B-Instruct.
- **The Qwen Series (Qwen et al., 2025).** include the effective Qwen2.5-7B-Instruct and Qwen2.5-72B-Instruct.
- **The GLM Series (GLM et al., 2024).** We use GLM4-Long to run the benchmark, since it is the commonly used in GLM series.
- **The Deepseek Series (Liu et al., 2024a; DeepSeek-AI et al., 2025).** We use Deepseek-V3 and Deepseek-R1 to run the benchmark.
- **The GPT Series (Achiam et al., 2023).** We evaluate GPT-4o and OpenAI o3-mini on our proposed benchmark.

We attempted to conduct experiments on various math-oriented LLMs, such as Metamath-Llemma-7B (Yu et al., 2023), Deepseek-Math-7B-instruct (Shao et al., 2024), InternLM2-Math-7B (Ying et al., 2024) and MAMmoTH-7B (Xiang Yue, 2023). However, these models fail during

Table 2: Evaluation of LLMs on numerical contextual retrieval, comparison, and summary tasks across number list, stock, and weather datasets. Also, * indicates that scores are calculated based on a short subset of outputs, as these models cannot handle long contexts and exhibit disruption when tested on longer instances.

Model	Retrieval			Comparison			Summary			Logic
	Number	Stock	Weather	Number	Stock	Weather	Number	Stock	Weather	Sequence
Random	12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5	12.5
Llama-3.1-8B-Inst	22.8	14.4	12.5	19.5	11.7	13.7	18.1	13.8	13.9*	18.2
Llama-3.1-70B-Inst	37.3	17.4	33.0	28.3	15.0	28.7	24.7	16.4	15.2	17.8
Llama-3.3-70B-Inst	44.4	19.4	32.6	31.5	13.8	35.8	26.3	16.8	18.0	18.6
Llama-3.1-405B-Inst	44.6	26.8	23.8	25.1	14.8	29.8	32.9	17.0	16.1	16.6
Llama-3.1-Nemotron-70B-Inst	41.6	19.3	33.5	26.6	13.7	33.6	29.4	16.5	17.0	16.4
Qwen2.5-7B-Inst	20.2	17.3	22.8	24.8	17.8	18.8	18.5	11.7	13.8	14.4
Qwen2.5-72B-Inst	28.8	41.4*	14.0*	28.0	26.0*	31.0*	31.9	18.8*	16.4*	19.0
GLM-4-Long	26.5	19.5	19.4	18.9	14.8	21.6	20.8	10.8	10.5	17.6
Deepseek-V3	47.2	47.5	27.6	27.0	22.5	35.8	21.8	13.0	15.1	15.8
GPT-4o	41.7	37.5	48.8	30.6	33.0	64.2	11.6	17.4	16.5	14.6
o3-mini	96.8	68.6	78.1	84.5	69.9	83.9	68.6	68.5	72.8	66.4
Deepseek-R1	73.6	81.3	93.6	64.5	85.3	98.8	62.1	64.3	61.7	65.4
Human Evaluation	100	52.6								

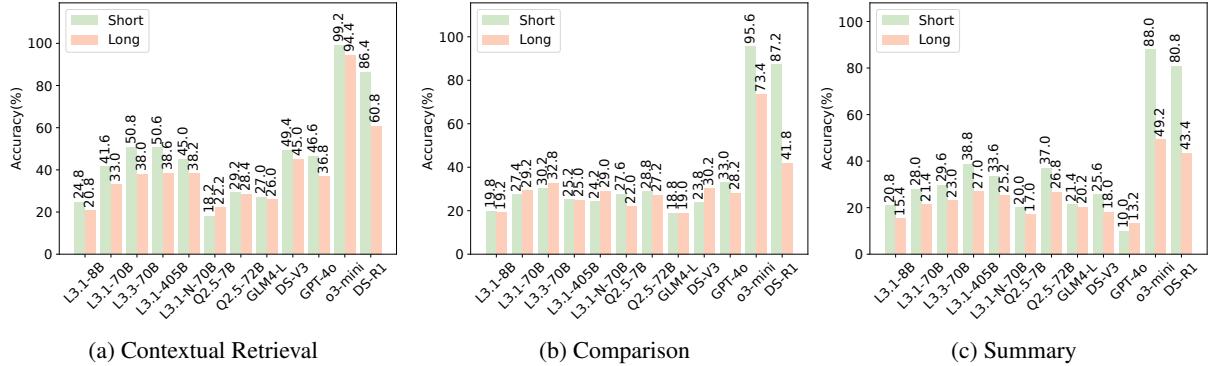


Figure 2: Evaluation on short and long context on number list.

experiments for various reasons such as overly long output sequence length and limited input sequence length. Fail cases are demonstrated in the Figure 11, Figure 12, Figure 13, and Figure 14 in Appendix.

4.2 Main Experiments

Evaluation on Contextual Retrieval, Comparison, Summary, and Logic Reasoning Abilities. As shown in Table 2, current popular and effective LLMs perform poorly on basic numerical tasks, including retrieval, comparison, summarization, and logic reasoning. The random baseline for each task is 12.5%, as there are 8 choices, and the probability of randomly selecting the correct answer is 1/8. Human evaluation was conducted by three undergraduate students.

Firstly, LLMs particularly struggle with accurately retrieving numerical data. This limitation arises from LLMs treating numbers as discrete tokens rather than continuous ones, coupled with insufficient exposure to structured numerical datasets during training, which restricts their ability to han-

dle simple numeric retrieval tasks. Most LLMs rely on surface-level patterns, treating numbers as symbols without understanding their true magnitudes. Studies like MATH-Perturb (Huang et al., 2025) and GSM-Symbolic (Mirzadeh et al., 2024) show that simple changes to numbers or variable names in math problems significantly reduce LLM accuracy, revealing their reliance on learned string patterns rather than genuine numerical understanding. Secondly, LLMs demonstrate weaknesses in recognizing numerical relationships, such as greater-than or less-than comparisons, due to a lack of numerical semantics and underdeveloped arithmetic reasoning capabilities. Thirdly, LLMs also perform poorly in summarizing numerical data (e.g., calculating sums or means), reflecting their inability to execute multi-step numerical operations. Similarly, logic reasoning tasks, especially those involving patterns or sequences, are particularly challenging, which are important for real-world applications. These tasks require multi-step reasoning, pattern recognition, and arithmetic operations, which ex-

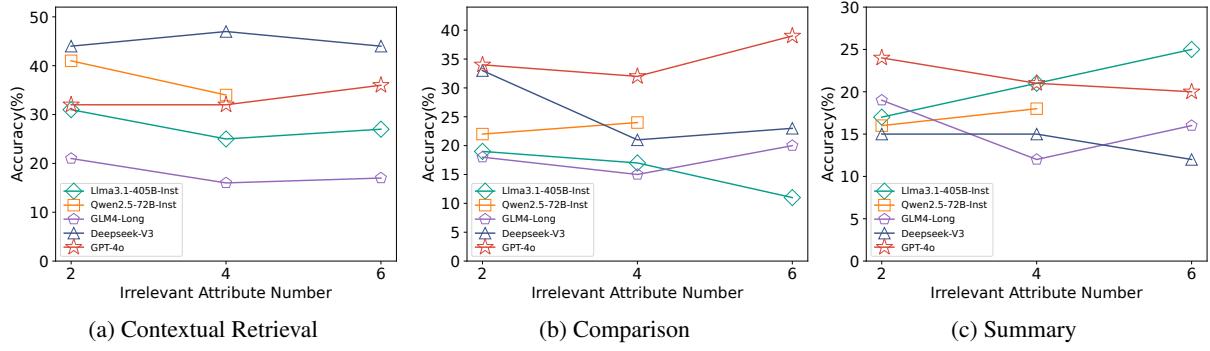


Figure 3: Evaluation on noisy stock dataset. Due to the input sequence length limit of Qwen2.5-72B-Inst on the API platform, the data containing 6 irrelevant attributes cannot be evaluated using this model.

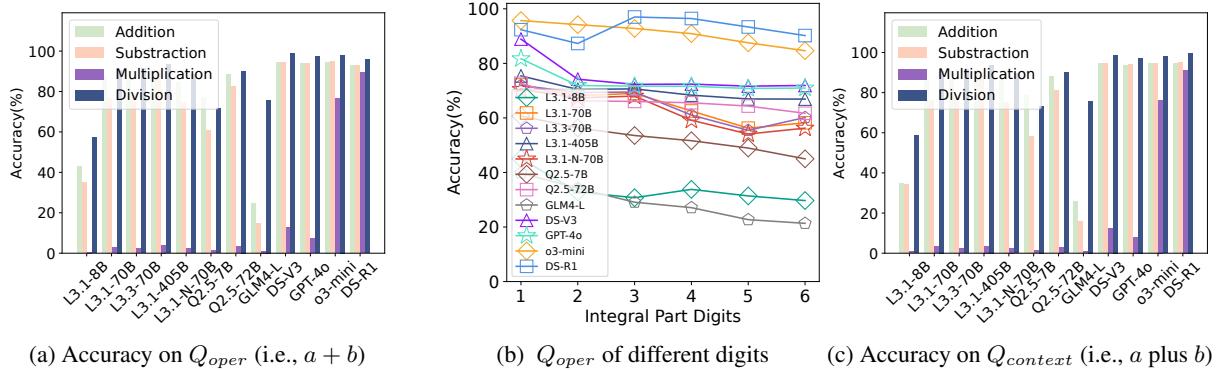


Figure 4: Evaluation on arithmetic operation.

pose the architectural limitations of current LLMs.

Evaluation on Different Context Length. We evaluate LLMs on varying context lengths. Specifically, we categorize the contexts of number lists, stock, and weather into short and long contexts. The average token numbers for the short and long contexts across the three datasets are listed in Table 11. As illustrated in Figure 2, Figure 8, and Figure 9, LLMs generally achieve lower accuracy on long contexts compared to short contexts. This is because long contexts require the model to have a stronger ability to capture long-range dependencies. Furthermore, if an LLM fails to perform well on short contexts, it is unlikely to achieve good results on long contexts. It highlights the importance of LLMs in understanding numeric data.

Evaluation on Noisy Context. To evaluate the numerical abilities of LLMs in noisy context, we augmented the structured data with unrelated attributes while preserving the original relevant data. For instance, in the stock dataset, we added attributes such as Stock Price-to-Earnings, Dividend Per Share (DPS), and 52-Week High/Low to each instance. These attributes are unrelated to the user queries, and the amount of noise corresponds to the number of irrelevant attributes added. Specifically, we introduced $k \in \{2, 4, 6\}$ irrelevant attributes

to each instance in both the stock and weather datasets. As shown in Figure 3 and Figure 10 in the Appendix, as k increases, most LLMs exhibit degraded performance. This demonstrates that irrelevant context can negatively impact the numerical retrieval and reasoning abilities of LLMs.

Evaluation on Arithmetic Operations Similarly, we evaluate five strong LLMs on arithmetic operations. Specifically, as illustrated in Figure 4 (a), even for simple arithmetic operations involving two numbers, LLMs fail to achieve 100% accuracy. Moreover, as the number of digits increases shown in Figure 4 (b), the accuracy of LLMs decreases, highlighting their limited ability to handle arithmetic tasks effectively, which is also observed in (Qiu et al., 2024). This poor performance stems from how LLMs generate responses. LLMs predict the highest-order digit before the lower-order digit (Zhang-Li et al., 2024), contradicting the standard arithmetic logic of progressing from lower- to higher-order digits. In particular, Figure 4 (a) and (c) shows that LLMs perform similarly on addition, subtraction, and division operations but achieve extremely low accuracy on multiplication tasks.

Evaluation on Number Recognition via Mixed-number-string Dataset. We evaluate the number recognition ability of effective LLMs by identify-

Table 3: Evaluation on mixed-number-string data with lengths ranging from 50 (i.e., 50 L) to 200.

Model	50 L	100 L	150 L	200 L
LLama3.1-405B	10.8	9.2	3.2	2.2
Qwen2.5-72B	3.0	1.2	0.6	0.2
GLM4-Long	6.6	4.8	3.0	2.4
GPT-4o	18.2	6.4	4.0	4.2
DeepSeek-V3	13.2	4.0	3.2	2.0
Human Eval	100	100	100	100

ing numbers from mixed-number-string sequences. For this evaluation, we select five effective LLMs based on Table 2, including DeepSeek-v3, GLM-4-Long, Llama3.1-405B, and Qwen2.5-72B. As shown in Table 3, all LLMs achieve extremely low accuracy in counting numbers within strings. Moreover, as the length of the string increases from 50 to 100, the accuracy of the LLMs decreases further. These results highlight that LLMs are significantly weak at distinguishing numbers from strings. The underlying reason is that current LLMs treat numbers as strings during training. This training paradigm inherently limits their ability to understand and process numbers effectively. Also, the Tokenizer can split a single number into multiple tokens, which can negatively affect the numeric meaning of each number.

Table 4: Chain of Thought Evaluations.

Model	Number List			Stock		
	R	C	S	R	C	S
Base	44.4	31.5	26.3	19.4	13.8	16.8
Plain-CoT	65.2	39.4	29.4	24.8	27.7	16.6
PS-CoT	65.4	40.0	26.4	24.3	16.7	15.8
Table-CoT	65.8	38.4	29.0	27.6	29.1	9.6

4.3 LLMs with Advanced Techniques

Chain of Thoughts. We evaluate the impact of Chain of Thought (CoT) on improving the numeric reasoning abilities of LLMs. We use Llama-3.3-70B-Instruct as the backbone LLM for evaluations. Specifically, we incorporate representative CoT approaches, including Plain CoT (Wei et al., 2023), Plan-and-Solve (PS)-CoT (Wang et al., 2023), and Table-CoT (Jin and Lu, 2023), to enhance the reasoning capabilities of Llama-3.3-70B-Instruct. As shown in the Table 4, these CoT techniques slightly improve LLM performance on NumericBench for simple tasks by refining outputs, but they significantly increase processing time due to longer outputs. However, CoT fails to enhance performance on complex stock summary tasks, where additional

Table 5: Few-shot Learning Evaluations.

Model	Number List			Stock		
	R	C	S	R	C	S
Base	44.4	31.5	26.3	19.4	13.8	16.8
One-shot	43.3	26.0	21.3	23.3	13.8	18.5
Two-shot	43.6	26.0	21.3	21.4	15.0	14.8

reasoning steps may introduce noise. This highlights the need to develop LLMs with stronger inherent numerical reasoning abilities.

Few-shot Learning. We evaluate the impact of few-shot demonstrations on improving the numeric reasoning abilities of LLMs. Specifically, we use Llama-3.3-70B-Instruct as the backbone and employ one-shot and two-shot setups. As shown in Table 5, the base model performs best on simpler tasks, such as Number List Retrieval, Comparison, and Summary. Few-shot learning (one-shot and two-shot setups) shows potential for improving performance in more complex tasks, such as Stock Retrieval and Summary. However, the performance gains from few-shot learning are inconsistent, suggesting that task complexity and the type of few-shot setup significantly influence the results.

Table 6: Supervised Fine-tuning.

Model	Number List			Stock			Weather		
	R	C	S	R	C	S	R	C	S
Base	24.4	21.1	29.5	14.0	9.3	16.8	17.0	10.8	11.1
QLoRA	62.8	52.3	59.1	14.0	9.3	15.0	9.3	14.5	9.1

Supervised Fine-tuning. We use QLoRA (Dettmers et al., 2023) to fine-tune Llama-3.1-8B-Instruct on our number list, weather, and stock datasets with shorter lengths. Each dataset is divided into 70% for training, 10% for evaluation, and 20% for testing. Table 6 summarizes our results on the test sets of the split datasets. As shown in Table 6, the fine-tuned model achieves improvements in retrieval, comparison, and summary tasks on the simple Number List dataset. However, for more complex real-world datasets, such as comparison and summary tasks on stock and weather data, the fine-tuning process does not yield improvements. This highlights the limitations of fine-tuning in enhancing LLM performance on complex, real-world data and underscores the necessity of developing LLMs with inherent numerical reasoning and understanding capabilities to handle such scenarios effectively.

4.4 Discussions on Numeracy Gaps of LLMs

In summary, extensive experimental results show that current state-of-the-art LLMs perform poorly on six fundamental numerical abilities. Here we discuss five potential reasons behind their poor performance on numerical tasks.

Tokenizer Limitation. LLMs use tokenizers to split input text into smaller units (tokens). Thus, Numbers are split into chunks as strings, based on statistical patterns in the training data. For example, 10000 is split into 100 and 00 tokens¹. These tokenizers do not consider the real meaning of numbers and continuous magnitude of numbers (Wallace et al., 2019; Singh and Strouse, 2024). Thus, LLMs do not perform well on simple number retrieval and comparison tasks. Recent studies (Sathe et al., 2024; Shen et al., 2023; Yang et al., 2025) demonstrate that diverse tokenization methods can enhance LLMs’ numerical understanding.

Training Corpora Limitation. LLMs are trained on extensive corpora, which also limits their ability to understand numerical-related symbols, such as *. For example, the multiplication of 246 and 369 can be denoted as 246*369. However, 246*369 may be interpreted as a password or encrypted text, since * in text strings is often associated with encryption. Consequently, enabling LLMs to accurately interpret arithmetic symbols and perform numerical reasoning remains an open problem, as their understanding of these symbols is heavily influenced by the statistical patterns and contexts encountered during training (Razeghi et al., 2022).

Training Paradigm Limitation. The training of LLMs relies on the next-token prediction paradigm, which is inherently misaligned with the logic of numerical computation. For example, when solving $16 + 56$ with the result being 72, an LLM will first predict the highest-order digit of the answer (i.e., 7) before predicting the lower-order digit (i.e., 2). This approach contradicts the fundamental logic of arithmetic computation, which typically proceeds from the lower-order digit to the higher-order digit. This discrepancy implies that LLMs effectively need to know the entire result upfront to generate digits sequentially in the correct order. As a result, LLMs struggle to perform well even on simple arithmetic operations.

Positional Embedding Limitation. Note that LLMs incorporate positional embeddings for tokens in sequence inputs. In arithmetic operations

like $12 + 26$ and $26 + 12$, the order of the numbers does not affect the result. However, LLMs assign different positional embeddings to the number 12 in each equation, as its position in the sequence differs. This lack of invariance in positional embeddings for numbers can influence the results. Therefore, how to design the positional embedding that improves numerical ability of LLMs without affecting the text understanding of LLMs is critical (McLeish et al., 2024; Golovneva et al., 2024).

Transformer Architecture Limitation. LLMs use Transformer to process input sequence, which rely on pattern recognition rather than explicit algorithmic reasoning. The computational power of transformers has upper bounds (Merrill and Sabharwal, 2023). Considering the complexity of arithmetic operations in real-world applications, it still needs to be theoretically investigated whether transformers can perform well on numerical operations.

5 Conclusion and Future Directions

In this paper, we propose a comprehensive benchmark NumericBench to evaluate the six fundamental numerical abilities of LLMs, including number recognition, arithmetic operations, contextual retrieval, comparison, summary, and logical reasoning. Our experiments reveal significant gaps in LLMs’ numerical reasoning, as even state-of-the-art models like GPT-4o and DeepSeek-V3 struggle with simple arithmetic, number retrieval, and multi-step reasoning tasks. These shortcomings arise from tokenization issues, training paradigms, and architectural limitations, underscoring the need for more numerically-aware modeling approaches.

To address these gaps, several future directions deserve exploration. First, developing numerically-aware tokenizers that treat numbers as continuous magnitudes can enable LLMs to better understand numerical concepts. Second, designing pretraining objectives specifically tailored to numerical reasoning, rather than relying solely on next-token prediction, can help models become more proficient at solving numerical problems. Third, incorporating structured numerical datasets during training can enhance real-world applicability by grounding models in accurate and practical numerical contexts. Finally, exploring suitable positional embeddings and hybrid symbolic-numeric architectures shows significant promise for improving the numerical capabilities of LLMs.

¹<https://gptforwork.com/tools/tokenizer>

Limitations

There are two main limitations of this paper. Firstly, the numerical tasks encountered in real-world scenarios are often far more complex and diverse compared to the six datasets proposed in NumericBench. Expanding the scope to include a broader range of numerical reasoning categories, such as traffic, would provide a more comprehensive assessment. Nevertheless, our work can serve as a meaningful point, highlighting the current limitations of LLMs in numerical tasks. We also analyze the potential reasons why LLMs struggle with numerical reasoning tasks, which can be attributed to the inherent limitations of transformer architectures and the next-token prediction objective. We hope it inspires further efforts to address these challenges and develop more advanced LLMs with enhanced numerical capabilities.

Secondly, although we evaluate twelve state-of-the-art LLMs, several newer LLMs and their variants, such as Claude and OpenAI o1 from major companies, are not included in our experiments. The reason for this exclusion is the expensive cost of accessing these model APIs. In brief, evaluating additional LLM variants across Claude, OpenAI, Mistral and GLM, typically requires a budget of nearly **\$15,000** US dollars. Specifically, experiments on the datasets in Table 2 require approximately 180 million tokens as inputs, while all left experiments (e.g., noisy contexts) require about 84 million tokens as inputs. For 1 million input tokens, Claude 3 Opus costs \$15², Claude 3.5 Sonnet costs \$3³, OpenAI o1 costs \$15⁴, Gemini 1.5 Pro costs \$12.5⁵, GLM4-Plus costs \$6.89⁶, Mistral Large 24.11 costs \$2⁷, and Mixtral 8x22B costs \$2⁸.

If we conduct experiments above with these top-tier models from major companies, it would cost at least 3960 dollars for Claude 3 Opus, 3960 dollars for OpenAI o1, 3300 dollars for Gemini 1.5 Pro, 1819 dollars for GLM4-Plus, 792 dollars for Claude Sonnet 3.5, 528 dollars for Mistral Large 24.11 and 528 dollars for Mixtral 8x22B, which is beyond our expected total experiment cost.

²<https://www.anthropic.com/pricing#anthropic-api>

³<https://www.anthropic.com/pricing#anthropic-api>

⁴<https://openai.com/api/pricing/>

⁵https://ai.google.dev/pricing#1_5pro

⁶<https://bigmodel.cn/pricing>

⁷<https://mistral.ai/en/products/la-plateforme>

⁸<https://mistral.ai/en/products/la-plateforme>

Also, for models such as OpenAI o1, which require generating really longer outputs for reasoning purposes, the output length is often unpredictable, while the model charges for \$60 per million output tokens, making the experiments even more expensive and difficult to control. Considering that GPT-4o, OpenAI o3-mini and DeepSeek-R1 represent the most state-of-the-art LLM models, we believe our evaluation can reflect the current numerical abilities of leading-edge LLMs. Therefore, our evaluation highlights the weaknesses of LLMs in numerical abilities and serves as a bridge to inspire further research focused on improving the numerical capabilities of these models.

Ethics Statement

This work does have any ethical issues.

Acknowledgments

Prof. Lei Chen’s work is supported by National Key Research and Development Program of China Grant No. 2023YFF0725100, National Science Foundation of China (NSFC) under Grant No. U22B2060, Guangdong-Hong Kong Technology Innovation Joint Funding Scheme Project No. 2024A0505040012, the Hong Kong RGC GRF Project 16213620, RIF Project R6020-19, AOE Project AoE/E-603/18, Theme-based project TRS T41-603/20R, CRF Project C2004-21G, Guangdong Province Science and Technology Plan Project 2023A0505030011, Guangzhou municipality big data intelligence key lab, 2023A03J0012, Hong Kong ITC ITF grants MHX/078/21 and PRP/004/22FX, Zhujiang scholar program 2021JC02X170, Microsoft Research Asia Collaborative Research Grant, HKUST-Webank joint research lab and 2023 HKUST Shenzhen-Hong Kong Collaborative Innovation Institute Green Sustainability Special Fund, from Shui On Xintiandi and the InnoSpace GBA. Prof. Qing Li is supported by the Hong Kong Research Grants Council under General Research Fund (project no. 15200023) and Research Impact Fund (project no. R1015-23). Dr. Haoyang Li is supported by research funds P0052504 and P0053707.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman,

- Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. *Preprint*, arXiv:2402.00157.
- Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*.
- Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.
- Gregor Bachmann and Vaishnav Nagarajan. 2024. The pitfalls of next-token prediction. *arXiv preprint arXiv:2403.06963*.
- Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhdian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.
- Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, et al. 2024. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks. *arXiv preprint arXiv:2412.15204*.
- Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.
- Xinran Chen, Xuanang Chen, Ben He, Tengfei Wen, and Le Sun. 2024. Analyze, generate and refine: Query expansion with llms for zero-shot open-domain qa. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11908–11922.
- Woon Sang Cho, Pengchuan Zhang, Yizhe Zhang, Xijun Li, Michel Galley, Chris Brockett, Mengdi Wang, and Jianfeng Gao. 2019. Towards coherent and cohesive long-form text generation. *Preprint*, arXiv:1811.00511.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.
- Eastmoney Website. 2025. Eastmoney Website. <https://www.eastmoney.com>. [Online; accessed 13-February-2025].
- Guhao Feng, Kai Yang, Yuntian Gu, Xinyue Ai, Shengjie Luo, Jiacheng Sun, Di He, Zhenguo Li, and Liwei Wang. 2024. How numerical precision affects mathematical reasoning capabilities of llms. *arXiv preprint arXiv:2410.13857*.
- Jiahui Gao, Renjie Pi, Jipeng Zhang, et al. 2025. G-LLaVA: Solving geometric problem with multi-modal large language model. In *The Thirteenth International Conference on Learning Representations*.
- Team GLM et al. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *Preprint*, arXiv:2406.12793.
- Olga Golovneva, Tianlu Wang, Jason Weston, and Sainbayar Sukhbaatar. 2024. Contextual position encoding: Learning to count what's important. *arXiv preprint arXiv:2405.18719*.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.
- Hangfeng He and Weijie J. Su. 2024. A law of next-token prediction in large language models. *Preprint*, arXiv:2408.13442.
- Kaixuan Huang, Jiacheng Guo, Zihao Li, Xiang Ji, Jiawei Ge, Wenzhe Li, Yingqing Guo, Tianle Cai, Hui Yuan, Runzhe Wang, et al. 2025. Mathperturb: Benchmarking llms' math reasoning abilities against hard perturbations. *arXiv preprint arXiv:2502.06453*.
- Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. 2023. Financebench: A new benchmark for financial question answering. *arXiv preprint arXiv:2311.11944*.
- Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. 2024. A survey on large language models for code generation. *arXiv preprint arXiv:2406.00515*.
- Ziqi Jin and Wei Lu. 2023. Tab-cot: Zero-shot tabular chain of thought. *Preprint*, arXiv:2305.17812.
- Haoyang Li, Yiming Li, Anxin Tian, Tianhao Tang, Zhanhao Xu, Xuejia Chen, Nicole Hu, Wei Dong, Qing Li, and Lei Chen. 2024a. A survey on large language model acceleration based on kv cache management. *arXiv preprint arXiv:2412.19442*.
- Qintong Li, Jiahui Gao, Sheng Wang, Renjie Pi, Xueliang Zhao, Chuan Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2024b. Forewarned is

- forearmed: Leveraging llms for data synthesis through failure-inducing exploration. *arXiv preprint arXiv:2410.16736*.
- Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, Xuchao Zhang, Tianjiao Zhao, Amit Panalkar, Dhagash Mehta, Stefano Pasquali, Wei Cheng, Haoyu Wang, Yanchi Liu, Zhengzhang Chen, Haifeng Chen, Chris White, Quanquan Gu, Jian Pei, Carl Yang, and Liang Zhao. 2024. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *Preprint*, arXiv:2305.18703.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.
- Hongwei Liu, Zilong Zheng, Yuxuan Qiao, Haodong Duan, Zhiwei Fei, Fengzhe Zhou, Wenwei Zhang, Songyang Zhang, Dahua Lin, and Kai Chen. 2024b. Mathbench: Evaluating the theory and application proficiency of llms with a hierarchical mathematics benchmark. *Findings of the Association for Computational Linguistics: ACL 2024*.
- Sean McLeish, Arpit Bansal, Alex Stein, Neel Jain, John Kirchenbauer, Brian R Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, Jonas Geiping, Avi Schwarzschild, et al. 2024. Transformers can do arithmetic with the right embeddings. *arXiv preprint arXiv:2405.17399*.
- William Merrill and Ashish Sabharwal. 2023. The parallelism tradeoff: Limitations of log-precision transformers. *Transactions of the Association for Computational Linguistics*, 11:531–545.
- Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.
- Open-Meteo. Free Open-Source Weather API | Open-Meteo.com — open-meteo.com. <https://open-meteo.com/>. [Online; accessed 13-February-2025].
- Luyu Qiu, Jianing Li, Chi Su, Chen Jason Zhang, and Lei Chen. 2024. Dissecting multiplication in transformers: Insights into llms. *arXiv preprint arXiv:2407.15360*.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.
- Yasaman Razeghi, Robert L. Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning. *Preprint*, arXiv:2202.07206.
- Ashutosh Sathe, Divyanshu Aggarwal, and Sunayana Sitaram. 2024. Improving self consistency in llms through probabilistic tokenization. *Preprint*, arXiv:2407.03678.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models.
- Ruoqi Shen, Sébastien Bubeck, Ronen Eldan, Yin Tat Lee, Yuanzhi Li, and Yi Zhang. 2023. Positional description matters for transformers arithmetic. *Preprint*, arXiv:2311.14737.
- Aaditya K. Singh and DJ Strouse. 2024. Tokenization counts: the impact of tokenization on arithmetic in frontier llms. *Preprint*, arXiv:2402.14903.
- Gemini Team, Petko Georgiev, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics.
- Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. 2019. Do NLP models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5307–5315, Hong Kong, China. Association for Computational Linguistics.
- Alex Wang. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy,

- and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.
- Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. **Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models.** *Preprint*, arXiv:2305.04091.
- Yubo Wang, Haoyang Li, Fei Teng, and Lei Chen. 2025. Graph-based retrieval augmented generation for dynamic few-shot text classification. *arXiv preprint arXiv:2501.02844*.
- Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. **Chain-of-thought prompting elicits reasoning in large language models.** *Preprint*, arXiv:2201.11903.
- Ge Zhang Yao Fu Wenhao Huang Huan Sun Yu Su Wenhua Chen Xiang Yue, Xingwei Qu. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*.
- Haotong Yang, Yi Hu, Shijia Kang, Zhouchen Lin, and Muhan Zhang. 2025. **Number cookbook: Number understanding of language models and how to improve it.** *Preprint*, arXiv:2411.03766.
- Qinyuan Ye. 2024. **Cross-task generalization abilities of large language models.** In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 4: Student Research Workshop)*, pages 255–262, Mexico City, Mexico. Association for Computational Linguistics.
- Huaiyuan Ying, Shuo Zhang, Linyang Li, Zhejian Zhou, Yunfan Shao, Zhaoye Fei, Yichuan Ma, Jiawei Hong, Kuikun Liu, Ziyi Wang, Yudong Wang, Zijian Wu, Shuaibin Li, Fengzhe Zhou, Hongwei Liu, Songyang Zhang, Wenwei Zhang, Hang Yan, Xipeng Qiu, Jiayu Wang, Kai Chen, and Dahua Lin. 2024. **Internlmath: Open math large language models toward verifiable reasoning.** *Preprint*, arXiv:2402.06332.
- Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhen-guo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.
- Kexin Zhang, Qingsong Wen, Chaoli Zhang, Rongyao Cai, Ming Jin, Yong Liu, James Y Zhang, Yuxuan Liang, Guansong Pang, Dongjin Song, et al. 2024. **Self-supervised learning for time series analysis: Taxonomy, progress, and prospects.** *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Daniel Zhang-Li, Nianyi Lin, Jifan Yu, Zheyuan Zhang, Zijun Yao, Xiaokang Zhang, Lei Hou, Jing Zhang, and Juanzi Li. 2024. Reverse that number! decoding order matters in arithmetic learning. *arXiv preprint arXiv:2403.05845*.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2024. **A survey of large language models.** *Preprint*, arXiv:2303.18223.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.
- Yongchao Zhou, Uri Alon, Xinyun Chen, Xuezhi Wang, Rishabh Agarwal, and Denny Zhou. Transformers can achieve length generalization but not robustly. In *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*.

A Appendix

In this appendix, we provide additional details about the design of **NumericBench**, along with supplementary experimental results and case studies. The organization of the supplementary materials in this appendix is as follows:

1. **Input format examples.** We transformed the structured JSON data into a string and appended it to the natural language question. The combined string, comprising the question and the structured data, was then used as the model’s input. In Table 7, we provide two examples.
2. **Question formats for contextual retrieval, comparison, and summary abilities.** As shown in Table 8, Table 9, and Table 10, we designed diverse question types tailored to each dataset to evaluate the three fundamental numerical abilities of LLMs: contextual retrieval, comparison, and summary. Contextual retrieval assesses the model’s capacity to accurately extract relevant numerical information from complex contexts; comparison tests the ability to analyze and compare numerical values; summary evaluates the synthesis of numerical information into concise and meaningful insights for tasks like reporting or trend analysis.
3. **Basic numerical questions answered incorrectly by GPT-4o.** As illustrated in Figure 5, Figure 6, and Figure 7, GPT-4o failed to answer three basic numerical questions correctly. This result is surprising, considering GPT-4o’s impressive performance in real-world applications. However, these findings highlight the weak fundamental numerical abilities of LLMs.
4. **Token counts for short and long contexts.** As shown in Table 11, the token counts of long and short contexts differ significantly. This distinction enables a more thorough evaluation of LLM performance across scenarios involving varying context lengths. Short contexts are designed to test the model’s ability to process and understand concise information, focusing on immediate comprehension and reasoning. In contrast, long contexts present a more complex challenge, requiring the model to handle extended sequences of information, maintain coherence over a larger context window, and retrieve relevant details from earlier parts of the input. Such

two type length can more comprehensively evaluate LLMs.

5. **Additional experimental results on noisy and varying-length contexts.** As shown in Figure 8 and Figure 9, existing LLMs perform poorly on the stock and weather datasets, although they achieve better performance compared to their results on short contexts. Similarly, as shown in Figure 10, LLMs perform poorly on noisy weather data.
6. **Output Token Statistics.** The number of tokens generated by LLMs is closely tied to their performance on NumericBench. Models augmented with CoT, as well as those inherently designed for reasoning, exhibit exceptional performance on our benchmark. However, this improvement comes with an increase in the number of output tokens. Table 12 provides a detailed analysis of the token counts for the outputs of these high-performing models.
7. **Real failure cases of math-oriented LLMs.** In this paper, we do not compare existing math-oriented LLMs, such as Metamath-Llemma-7B (Yu et al., 2023), Deepseek-Math-7B-Instruct (Shao et al., 2024), InternLM2-Math-7B (Ying et al., 2024), and MAMmoTH-7B (Xiang Yue, 2023). This is primarily because these math-oriented LLMs are designed for specialized geometric and structured mathematical problems. They are unable to understand the tasks in NumericBench, fail to follow a correct reasoning process, and directly produce meaningless outputs. These failure cases are illustrated in Figure 11, Figure 12, Figure 13, and Figure 14.

The Use of AI Tools. When writing this paper, we use Grammarly⁹ for automated spell checking and use GPT-4o¹⁰ to refine several sentences.

⁹<https://www.grammarly.com/>

¹⁰<https://platform.openai.com/docs/models/gpt-4o>

Table 7: NumericBench Input Format. Due to the excessive length of the dataset input, the input examples provided in the table use ... to indicate the omission of certain parts.

Dataset	Input Example
Number List	<p>You're an assistant designed to answer multiple choice questions. You'll be given some context and multiple choice questions about the context. For each question, you will only output the answer with the following format, without additional information such as how you solve the problem: The answer is (correct option). Option refers to capital letters like A, B, C, D, etc. Note that there is only one correct option.</p> <p>The data you receive will contain a list of numbers.</p> <p>Question: Which index holds the greatest number in the list between the indices 3 and 14?</p> <p>Options: A: 3, B: 4, C: 6, D: 7, E: 9, F: 10, G: 12, H: 13</p> <p>Data: <code>[-778.124, -384366.23, -856, -243, -961279, -10, ...]</code></p>
Stock	<p>You're an assistant designed to answer multiple choice questions. You'll be given some context and multiple choice questions about the context. For each question, you will only output the answer with the following format, without additional information such as how you solve the problem: The answer is (correct option). Option refers to capital letters like A, B, C, D, etc. Note that there is only one correct option.</p> <p>This dataset contains key financial and trading information for stocks. The 'stock_code' column represents the unique identifier for each stock, while 'date' indicates the trading date. 'close_price' and 'open_price' provide the close and open prices of the stock on the given trading day, respectively. 'quantity_relative_ratio' compares the day's trading volume to the average volume over a prior period ...</p> <p>Note that exceeding some number means greater than that number, while reaching some number means greater than or equal to that number.</p> <p>Question: How many days had a close price higher than the open price, with the quantity relative ratio exceeding 54%?</p> <p>Data: <code>[{date : 2023.11.03, close_price : 10.87, open_price : 10.9, quantity_relratio : 0.63, ...}, ...]</code></p>

Table 8: Question format on number list dataset. R: contextual retrieval, C: comparison, S: summary. In the contextual retrieval task, a number x is randomly selected from the given number list. For the comparison task, the k -th largest number is randomly generated within the range of one to the length of the number list. The indices x corresponds to twenty percent of the length of the number list, while y corresponds to eighty percent of the length. The number z is randomly chosen within the range $(\min(\text{list}), \max(\text{list}))$. For the summary task, the top k is set to thirty percent of the length of the number list.

Ability	Question Format
R	Q_0 : What is the index of the first occurrence of the number x in the list? Q_1 : What is the index of the last occurrence of the number x in the list? Q_2 : What is the number after the first occurrence of the number x in the list? Q_3 : What is the number before the last occurrence of the number x in the list? Q_4 : What is the index of the first even number in the list? Q_5 : What is the index of the first odd number in the list? Q_6 : What is the index of the last even number in the list? Q_7 : What is the index of the last odd number in the list?
C	Q_8 : What is the index of the first occurrence of the k -th largest number in the given list? Q_9 : Which index holds the greatest number in the list between the indices x and y ? Q_{10} : Which index holds the smallest number in the list between the indices x and y ? Q_{11} : Which number is closest to z in the list between the indices x and y ? Q_{12} : Which number is furthest from z in the list between the indices x and y ? Q_{13} : Which number is the largest among those less than z in the list? Q_{14} : Which number is the smallest among those greater than z in the list?
S	Q_{15} : What is the maximum sum of any two consecutive items in the list? Q_{16} : What is the maximum sum of any three consecutive items in the list? Q_{17} : What is the maximum absolute difference between two consecutive items in the list? Q_{18} : What is the sum of the indices of the top k largest numbers in the list? Q_{19} : What is the sum of the indices of the top k smallest numbers in the list? Q_{20} : What is the average of the indices of the top k largest numbers in the list? Q_{21} : What is the average of the indices of the top k smallest numbers in the list? Q_{22} : How many times do numbers consecutively increase for more than five times? Q_{23} : How many times do numbers consecutively decrease for more than five times?

Table 9: Question format on stock dataset. R: contextual retrieval, C: comparison, S: summary. x and y lie within the minimum and maximum ranges of their respective attributes. The top k corresponds to thirty percent of the number list. $date_1$ represents the day at the twentieth percentile of the stock history, while $date_2$ corresponds to the day at the eightieth percentile.

Ability	Question Format
R	Q_0 : On which date did the close price of the stock first reach x yuan?
	Q_1 : On which date did the highest price of the stock first reach x yuan?
	Q_2 : On which date did the volume of the stock first reach x lots?
	Q_3 : On which date did the value of the stock first reach x thousand yuan?
	Q_4 : On which date did the price change rate of the stock first reach $x\%$?
	Q_5 : On which date did the price change of the stock first reach x yuan?
C	Q_6 : On which date did the stock have the highest turnover rate when the close price was greater than x yuan?
	Q_7 : On which date did the stock have the highest quantity relative ratio when the open price was less than x yuan?
	Q_8 : On which date did the stock have the highest difference between the highest and lowest prices when the trading volume exceeded x lots?
	Q_9 : On which date did the stock record the highest daily average price, calculated as 'value' divided by 'volume,' when the PE ratio was less than x ?
	Q_{10} : Among the top- k trading value days, on which date did the stock have the lowest close price?
	Q_{11} : When the quantity relative ratio exceeded x , on which date did the stock have the highest sum of the open price and close price?
	Q_{12} : When the absolute price change rate exceeded $x\%$, on which date did the stock have the highest difference between the highest and lowest prices?
	Q_{13} : How many days had a volume greater than x from $date_1$ to $date_2$?
	Q_{14} : How many days had the close price higher than the open price from $date_1$ to $date_2$?
	Q_{15} : How many days had a close price higher than the open price, with the quantity relative ratio exceeding $x\%$?
	Q_{16} : How many days had the close price reach x yuan with the absolute price change rate exceeding $x\%$?
	Q_{17} : What was the average trading volume when both the turnover rate exceeded $x\%$ and the price change rate was greater than $y\%$?
S	Q_{18} : Excluding non-trading days, how many times did the open price of the stock rise for three or more consecutive days?
	Q_{19} : Excluding non-trading days, how many times did the close price of the stock rise for three or more consecutive days?
	Q_{20} : Excluding non-trading days, how many times did the open price and close price of the stock both rise for three or more consecutive days?

Table 10: Question format on weather dataset. R: contextual retrieval, C: comparison, S: summary. The value of x falls within the minimum and maximum ranges of its respective attribute. $date_1$ represents the day at the twentieth percentile of the stock history, while $date_2$ represents the day at the eightieth percentile.

Ability	Question Format
R	Q_0 : On which date did the temperature at two meters lastly reach $x^{\circ}\text{C}$? Q_1 : On which date did the relative humidity at two meters lastly exceed $x\%$? Q_2 : On which date did the dew point temperature at two meters lastly drop below $x^{\circ}\text{C}$? Q_3 : On which date did the precipitation lastly exceed $x \text{ mm}$? Q_4 : On which date did the sea-level air pressure lastly exceed $x \text{ hPa}$? Q_5 : On which date did the cloud cover lastly reach $x\%$? Q_6 : On which date did the wind speed at 10 meters lastly exceed $x \text{ m/s}$?
C	Q_7 : On which date did the temperature at two meters reach its highest value when the relative humidity was below $x\%$? Q_8 : On which date did the relative humidity at two meters reach its lowest value when the temperature at two meters was above $x^{\circ}\text{C}$? Q_9 : On which date did the difference between the temperature and dew point at two meters reach its maximum when the cloud cover was below $x\%$? Q_{10} : On which date did the precipitation reach its highest value when the temperature at two meters was below $x^{\circ}\text{C}$? Q_{11} : On which date did the cloud cover reach its lowest value when the wind speed at 10 meters exceeded $x \text{ m/s}$? Q_{12} : On which date did the wind speed at 10 meters reach its highest value when the precipitation exceeded $x \text{ mm}$? Q_{13} : On which date did the sea-level air pressure reach its highest value when the cloud cover was below $x\%$?
S	Q_{14} : How many days had a temperature at two meters greater than $x^{\circ}\text{C}$ from $date_1$ to $date_2$? Q_{15} : How many days had a relative humidity at two meters exceeding $x\%$ from $date_1$ to $date_2$? Q_{16} : How many days had a precipitation greater than $x \text{ mm}$ from $date_1$ to $date_2$? Q_{17} : What was the average temperature at two meters when the relative humidity exceeded $x\%$? Q_{18} : What was the average wind speed at 10 meters when the precipitation exceeded $x \text{ mm}$? Q_{19} : How many times did the temperature at two meters rise for three or more consecutive days? Q_{20} : How many times did the relative humidity at two meters drop for three or more consecutive days?

Which is greater, -9.8 or -9.11?

GPT-4o Poe

To determine which number is greater between **-9.8** and **-9.11**, remember that on the number line, larger negative numbers are *closer to zero*.

- **-9.8** is closer to zero than **-9.11**.
- Therefore, **-9.8** is greater than **-9.11**.

Figure 5: Number comparisons on GPT-4o. The correct answer is -9.11.

... To calculate 132.913×769.256

GPT-4o Poe

The product of **132.913 × 769.256** is calculated as follows:

Solution:

$132.913 \times 769.256 = 102,209.552528$

So, the result is **102,209.55** (rounded to two decimal places).

Figure 6: Number multiplication on GPT-4o. The correct answer is 102244.12.

How many times does 9.12 appear in the list? [123, 454, 4, 23, 234, 32.12, 9.12, 234, -12.452, 8.12, -123, -324, 193.2, 9.12, -9.12, 9.12, -132, 9.12, 213, 9.1203, 12.4]

GPT-4o Poe

3

Figure 7: Number counting on GPT-4o, which is required directly give answer. The correct answer is 4.

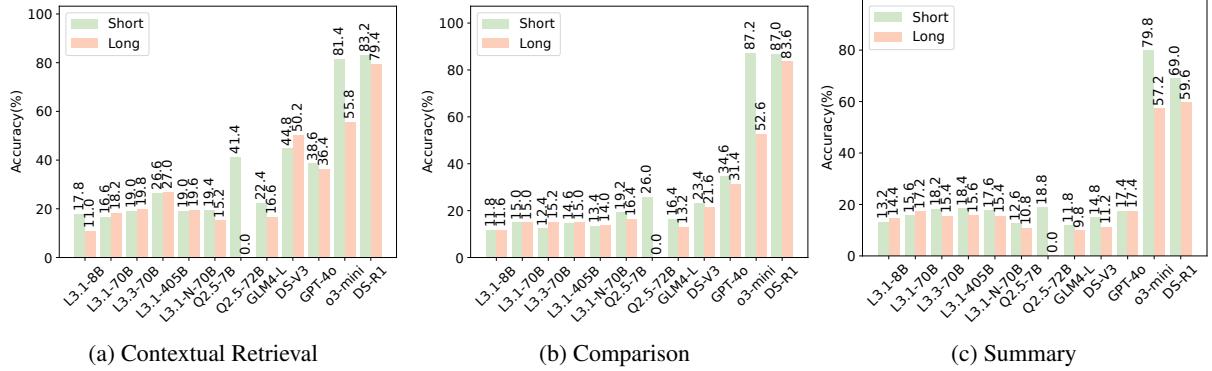


Figure 8: Evaluation on short and long context on stock dataset. Due to the input sequence length limit of Qwen2.5-72B-Inst on the API platform, the long dataset of all three abilities cannot be evaluated using this model.

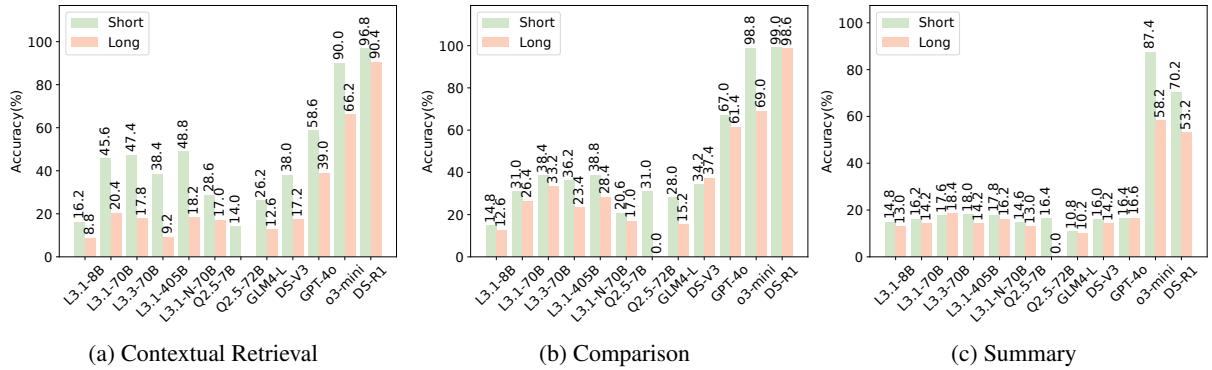


Figure 9: Evaluation on short and long context on weather dataset. Due to the input sequence length limit of Qwen2.5-72B-Inst on the API platform, the long dataset of all three abilities cannot be evaluated using this model.

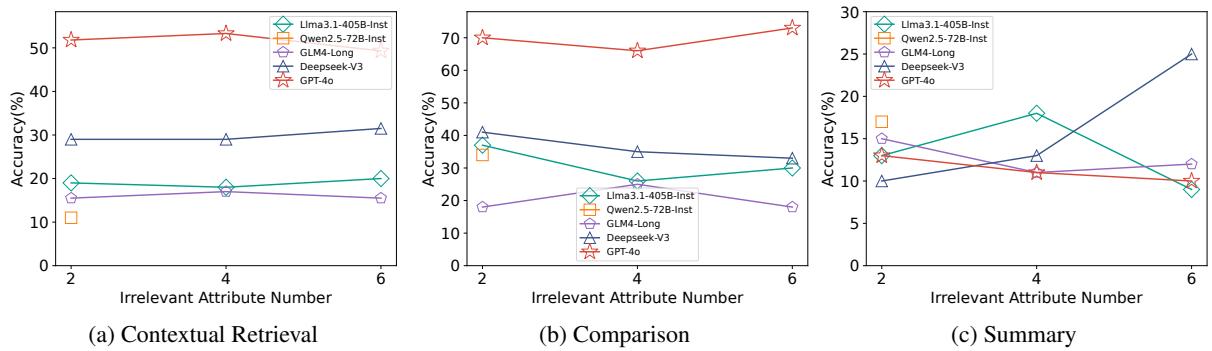


Figure 10: Evaluation on noisy weather dataset. Due to the input sequence length limit of Qwen2.5-72B-Inst on the API platform, the data containing 4 and 6 irrelevant attributes cannot be evaluated using this model.

Table 11: The average token number on short and long instances for each data.

Dataset	Ability	Short		Long	
		# Instance	Avg Token	# Instance	Avg Token
Number List	<i>Contextual Retrieval</i>	500	809.12	500	6599.34
	<i>Comparison</i>	500	804.86	500	6566.27
	<i>Summary</i>	500	822.49	500	6487.07
Stock	<i>Contextual Retrieval</i>	500	18529.07	500	36641.63
	<i>Comparison</i>	500	18539.58	500	36651.22
	<i>Summary</i>	500	18504.51	500	36618.07
Weather	<i>Contextual Retrieval</i>	500	18362.38	500	36356.13
	<i>Comparison</i>	500	18371.11	500	36365.27
	<i>Summary</i>	500	18334.15	500	36328.27

Table 12: Output Token Statistics. We utilize Llama-3.3-70B-Instruct as our base model.

Model	Number List		Stock		Weather		Sequence
	Short	Long	Short	Long	Short	Long	
Base	5.00	5.00	5.00	5.00	5.00	5.09	5.00
Plain-CoT	762.65	1386.60	456.45	402.18	-	-	1048.14
PS-CoT	928.01	2776.77	512.37	515.25	-	-	913.09
Table-CoT	872.65	1426.27	495.29	478.47	-	-	981.95
o3-mini	2966.66	4158.25	3898.36	3907.98	2655.78	3466.44	4400.85
Deepseek-R1	5427.32	10463.43	4559.96	4837.71	2228.91	2482.29	8057.13

Example A.1: InternLM2-Math-7B

Failure:

Model repeatedly output the same number until the output sequence ends.

User:

Question: What is the maximum sum of any three consecutive items in the list? Options: A: 21501.9737231253, B: 21502.8144848529, C: 21501.9642201794, D: 21502.7354388437, E: 21502.0327435643, F: 21501.7526575621, G: 21501.7057960155, H: 21502 Data: [-5279,(100 numbers)]

InternLM2-Math-7B Model:

Step 1: To find the maximum sum of any three consecutive items in the list, we need to identify the three largest numbers in the list and add them together. <PRM +>

Step 2: We can start by sorting the list in ascending order to make it easier to identify the largest numbers. <PRM +>

Figure 11: Failure type and conversation record of InternLM2-Math-7B. The model constantly generate the same "-8509" text until the output sequence was too long and truncated, leaving the problem unsolved.

Example A.2: DeepSeek-Math-7B

Failure:

Model outputs sequence without meaning, probably due to overly long input sequences.

User:

Question: What is the sum of the index of top 300 smallest numbers in the list? Options: A: 146734, B: 146738, C: 146733, D: 146724, E: 146728, F: 146727, G: 146721, H: 146729 Data: [3640, 1560, ... (100 numbers)]

Deepseek-Math-7B Model:

31,3,9,3,1,1,3,3,3,3,3,3,3,3, 3,3,1,3,3,3,1,3,3,3, 3, 3,3,3, 1, 1,3,3, 3,1,3, 3,3,3, 1, 3,3,3, 3,3,1,3,1,3,91, 3,3,3, 3,31,3,3,3, 3, 339, 3, 3, 311, 1, 3, 1,3,3, 3,3,6,3,3, 3,31,3,3,1, 1, 1, 1, 3,1,1, 3,3, 1,1, 3,1,3,1, 1, 1, 1, 1, 19,1, 1, 1,191,1, 1,1, 19, 31,1,1, 1, 3,1, 1,3, 1133,3, 1, 1,1, 11, 11,3, 31, 3, 3,1,1,3,1, 1,33, 11,3,1911,3, 1,13,3131311119191,1, 3,1,11,1,19,11,1,1,1....(keeps outputting numbers)

Figure 12: Failure type and conversation record of DeepSeek-Math-7B. The model only generates a series of numbers throughout its output, and we’re unable to acquire any useful information from the output.

Example A.3: Metamath-Llemma-7B

Failure:

Model outputs ineffective chain of thoughts, and giving the wrong answer in the wrong format(ground truth is B))

User:

Question: What is the sum of the index of top 30 smallest numbers in the list? Options: A: 1566, B: 1572, C: 1575, D: 1573, E: 1577, F: 1564, G: 1569, H: 1567 Data: [-2568.1542995196, 1035.3645340299...(100 numbers)]

Metamath-Llemma-7B Model:

The data you receive is a list of numbers. To find the sum of the index of the top 30 smallest numbers, we need to find the 30 smallest numbers in the list and then calculate their sum. To find the 30 smallest numbers, we can use the min function in Python. The min function takes a list as input and returns the smallest number in the list. We can use the min function to find the 30 smallest numbers in the list. Now, we need to calculate the sum of these 30 smallest numbers. To do this, we can use the sum function in Python. The sum function takes a list as input and returns the sum of all the numbers in the list. We can use the sum function to calculate the sum of the 30 smallest numbers. Finally, we need to find the index of the sum. To do this, we can use the index function in Python. The index function takes a number as input and returns the index of that number in the list. We can use the index function to find the index of the sum. Therefore, the answer is (correct option). ##### 1566 The answer is: 1566

Figure 13: Failure type and conversation record of Metamath-Llemma-7B. The model attempts to use programming language to solve the problem, but it does not provide executable lines of code, only providing vague description of certain functions instead. After outputting the vague solution, the model outputs an answer, which is different from the ground truth and not in the expected format(We expect the model to provide the answer with a capital letter corresponding to the correct option).

Example A.4: MAmmoTH-13B

Failure:

Model outputs ineffective chain of thoughts, and giving the wrong answer in the wrong format(ground truth is B))

User:

Question: What is the sum of the index of top 30 smallest numbers in the list? Options: A: 1566, B: 1572, C: 1575, D: 1573, E: 1577, F: 1564, G: 1569, H: 1567 Data: [-2568.1542995196, 1035.3645340299...(100 numbers)]

MAmmoTH-13B Model:

Notice that there are two numbers in your list that have the smallest index: 4346.5174909575 and 8169.6394989672. They have the smallest index of 4346 and 8169 respectively. Since in the options the first number is the index of the smallest number, then the correct option is D.

Option: D

Figure 14: Failure type and conversation record of Mammoth-13B. The model fail to understand the problem's instruction, and does not provide reasonable output. Moreover, it provides the wrong answer in incorrect format, just like Metamath-Llemma-7B(We expect the model to provide the answer with a capital letter corresponding to the correct option).