Frequency and Representation - Based Exploration of Mathematical Reasoning in Large Language Models

Ziyao Mou, Yiran Zhong, Elena Kote
Johns Hopkins University
zmou1@jhu.edu, yzhong43@jhu.edu, ekote1@jhu.edu

Abstract

Large language models (LLMs) have demonstrated impressive capabilities across a range of natural language tasks, yet their numerical reasoning abilities remain underexplored. In this work, we investigate the extent to which LLMs rely on the frequency and representation of mathematical formulas and numbers in their training data to perform arithmetic tasks. Using the Infini-gram framework, we analyze the occurrence frequencies of numbers and mathematical expressions in pre-training data and their impact on model accuracy. Extending existing research, which highlights a correlation between number frequency and performance in addition and multiplication, we delve into more complex operations, such as division, exponentiation, and logarithms. Lastly, we compare the influence of formula frequency on model accuracy, providing insight into the mechanisms underlying the numerical proficiency of LLMs. Our findings provide insights into the relationship between training data characteristics and arithmetic task performance, revealing both the strengths and limitations of LLMs in applying mathematical knowledge.

1 Introduction and Motivation

LLMs have achieved remarkable success in NLP tasks but often struggle with numerical reasoning, a key skill for a variety of applications. Current research indicates that their arithmetic accuracy is influenced by the frequency of numbers and mathematical expressions in training data, particularly for basic operations like addition and multiplication. However, the mechanisms driving these capabilities, especially for complex tasks like division and logarithms, remain unclear.

This study investigates how LLMs leverage the frequency and representation of numbers and operators in their training data. Using the Infini-gram framework, we retrieve the frequency of any number, word or symbol in the model's pre-training data . We then test the LLaMA2-7B model on a custom-made dataset, with the aim of exploring a comprehensive range of mathematical operations and evaluate the model's accuracy by comparing its responses to the dataset's ground truth answers.

2 Related Work

2.1 Infini-gram: Scaling Unbounded n-gram Language Models to a Trillion Tokens

Liu et al. [2024] introduces their modernization on n-gram LMs in two key aspects. First, they trained the models on the same data scale as neural LLMs, utilizing 5 trillion tokens—making this the largest n-gram LM ever constructed. Additionally, they developed a system called "infini-gram", powered by suffix arrays (the technique support fast queries), which is capable of computing ∞-gram (as well as n-gram with arbitrary values of n) probabilities with millisecond-level latency. The ∞-gram LM demonstrates high accuracy in next-token prediction (47%) and can complement neural LLMs to significantly reduce perplexity. For our project, this technique could help us analyze the frequency of numbers and mathematical expressions in the LLM training data, offering insight into whether the model is recalling memorized patterns or reasoning through problems.

2.2 Mathemtatical Reasoning in LLMs

Research has shown that LLMs can handle basic arithmetic operations, but their proficiency in more complex tasks such as algebra, calculus, and multi-step problem-solving remains limited. Studies such as those by Frieder et al. [2023] have compared different versions of models like ChatGPT and GPT-4, focusing on their mathematical problem-solving capabilities, including theorem proving and computation. These investigations highlight the need for more nuanced evaluations of LLMs' mathematical abilities, particularly when it comes to generating accurate proofs and performing multi-step reasoning. Mirzadeh et al. [2024] introduced GSM-Symbolic, a benchmark designed to provide more controlled and diverse evaluations of LLMs' mathematical reasoning abilities. Their findings reveal significant limitations in LLMs' reasoning capabilities, especially when the numerical values in questions are altered or when additional clauses are introduced, even if irrelevant to the reasoning chain. Works by Chang et al. [2023] and Testolin [2023] explore how LLMs perform in various mathematical domains, revealing challenges in tasks requiring deep logical reasoning. A key observation is that LLMs often struggle with tasks that involve multiple steps or require abstract reasoning, suggesting that their current capabilities may be more reflective of memorization from training data rather than true generalization.

3 Methodology and Techniques

3.1 Model

For generating answers and conducting the subsequent analysis, we used LLaMA2-7B, a casual language model developed by Meta. We selected this model because it has a robust architecture and a relatively moderate parameter size, providing a balance between computational efficiency and generalizability.

3.2 Dataset

We generated a dataset with the aim of exploring a comprehensive range of mathematical operations with varying levels of complexity. Specifically, we looked at addition, subtraction, multiplication, division, exponentiation, and logarithms. We tested in the following inclusive ranges of numbers:

[10, 19], [50,59], [100, 109], [250, 259], [500, 509], [1000,1009], [2500, 2509], [5000, 5009], [10000, 10009].

For each number and operation, we generated questions in multiple possible representations. For numbers, the representations we explored were the following:

1. **Symbolic**: Numerical representation for both operands: "10 + 5"
2. **Text 1**: Text and numeric representation: "ten + 5"
3. **Text 2**: Numeric and text representation: "10 + five"
4. **Text 3**: Text representation for both operands: "ten + five"

For each operator, we used different representations that we tested the model performance on, as shown below:

1. **Addition**: +, plus, add
2. **Subtraction**: -, minus, subtract
3. **Multiplication**: *, times, multiply
4. **Division**: /, divided by, over
5. **Exponentiation**: ^, to the power of, **
6. **Logarithm**: log base

As a result of all these possible representations, for each number, our dataset has a total of 640 questions on arithmetic operations containing that number as the first operand. Across all numbers in the ranges that we explored, our dataset has a total of 57600 arithmetic questions. To help with the calculation of accuracy later on, we also included ground truth values to the dataset.

3.3 Counting Tool

We used infini-gram(Liu et al. [2024]) as our counting tool.We utilized the web API provided by the authors, as downloading the entire index of the LLaMA-2 pre-training data they constructed would be cost-prohibitive. Our parallel calls to the web API proved to be efficient, allowing us to complete the study in a timely manner.

3.4 Experiment Setting

We utilized the LLaMA2-7B model for inference, running experiments on A100 GPU. Frequency computation was performed using the index of `v4_dolma-v1_7_llama`, constructed through the Infini-gram framework.The study was conducted using the custom build dataset and the results from model inference were compared against the dataset's ground truth values after regularization, to obtain model accuracy. To evaluate correctness, model predictions were required to achieve a precision of 10^{-7}.

4 Results and Analysis

4.1 Analysis of Number and Formula Frequency in Pretraining Data

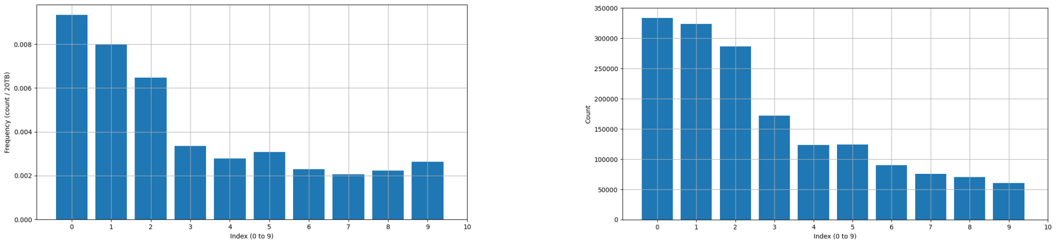
Using Infinigram's counting method, we analyzed the frequency of numbers and formulas in the LLaMA-2 7B pre-training data. Specifically, we investigated the frequencies of single-digit and two-digit numbers as well as single-digit and two-digit formulas. Additionally, we examined the occurrences of different representations of numbers (numeric vs. literal) and symbols within formulas. We split our analysis in terms of single-digit and two-digit numbers because single-digit numbers are substrings of two-digit numbers, so we would naturally expect their counts to be much higher than those of two-digit numbers. Similarly with numbers expressed with words (ie fifty-five instead of 55), single-digit number words are substrings of two-digit number words (with the exception of numbers from 10 to 20).

It is important to note that the counts we are getting from Infinigram for every string/number include the counts of that string or symbol in the entire dataset, regardless of the context it was used in. This explains some of the trends we will see in the following, where, for example, the symbol for subtraction (-) has a very high count, because that symbol can be used in many contexts that are not necessarily mathematical, whereas a symbol such as ˆ would naturally have a much lower count due to far fewer use-cases.

4.1.1 Frequency Analysis of Single-Digit Numbers and Formulas

Initially, we quantified the occurrences of numbers in the entire dataset using the Infini-gram counting method. As illustrated in Figure 1a, the digit 0 exhibits the highest frequency among all numbers, with a general decreasing trend observed from 0 to 9. We also analyzed the generated data to quantify the frequency of each digit's occurrence within formulas, accounting for various representations of both numbers and symbols. The results, shown in Figure1b, demonstrate a consistent decreasing trend in frequency from 0 to 9.

As shown in Figure 2, We also analyzed the frequency of different one-digit calculation types, observing that calculation frequency decreases from 0 to 9. It is worth noting, however, that division calculations include occurrences of the / symbol, which counts not only fractional expressions but also strings such as dates. Similarly, subtraction counts include equations containing the - symbol. Since it is challenging to distinguish between operator and non-operator contexts, our subsequent single-operator analysis will focus on addition and multiplication only. We are also concerned that, apart from division, 1 and 2 are associated with the most one-digit calculations, which is also consistent with our general knowledge.



(a) Frequency of occurrence of numbers in all LLaMA-2-7B pre-training data

(b) Counts of one-digit (numeric and text) occurrences in formulas for all LLaMA-2-7B pre-training data

Figure 1: Comparative analysis of number frequency and one-digit occurrence counts in LLaMA-2-7B pre-training data

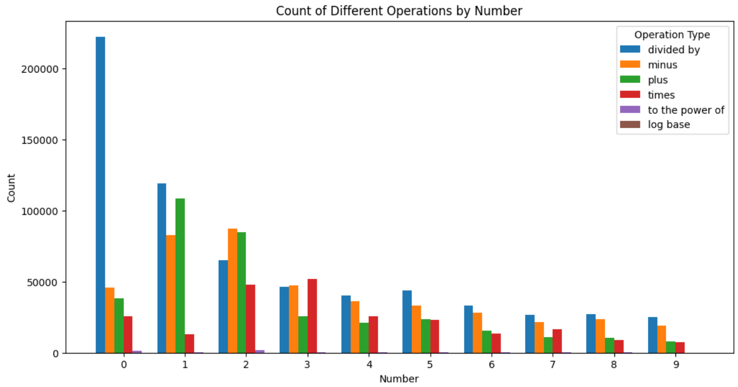


Figure 2: Count of different operations by 1-digit Number in all LLaMA-2-7B pre-training data

4.1.2 Frequency Analysis of Multi-Digit Numbers and Formulas

As shown in the Figure3, we examined the frequency of one-, two-, and three-digit multiplication occurrences in the dataset. Our analysis revealed that computations with higher digit counts appear less frequently in the pre-training data. Furthermore, Deng et al. [2024] indicates a sharp decline in chatGPT4-o accuracy for multiplication tasks of two and three digits. Consequently, we will focus our study on one- and two-digit numerical formulas.

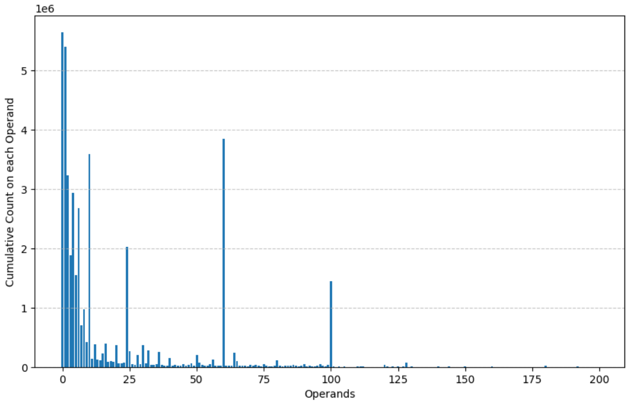


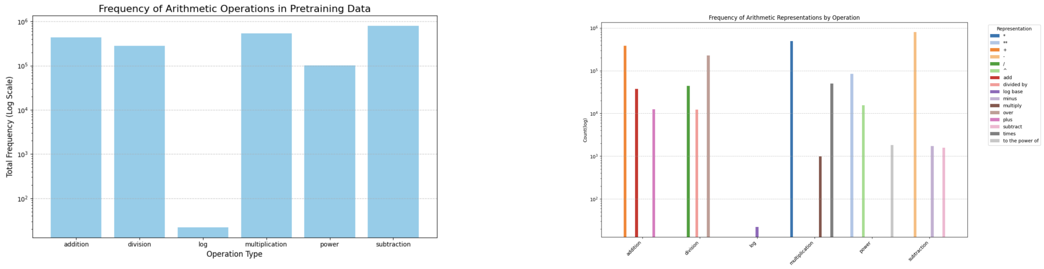
Figure 3: Count of operand multiplication equations up to 200 in LLaMA-2-7B pretraining data

4.2 Frequency Analysis of Operators and Mathematical Symbols

The arithmetic operations on which we focused our analysis were addition, subtraction, multiplication, division, power/exponentiation, and logarithms. We looked at different representations of each of these operators, all of which are listed in Table 1 . The goal of our analysis was to see how a model's accuracy changes for different operators, as well as how a model's accuracy varies on different representations of the same arithmetic operation. We think variations of the latter type could be an indicator of the fact that the model is memorizing results, as opposed to learning the logic behind arithmetic operations. In other words, if a model actually understands the logic behind adding two two-digit integers, then it should be able to answer an addition question correctly regardless of what is used to represent that addition. Figure4 displays the counts of each of the operators and their representations in the training data. As briefly mentioned before in the preliminary analysis, some operator representations have a higher count, not only because some arithmetic operations might appear more frequently in the training data, but also because their symbols could often be used outside of a mathematical context. For example, '*' can often be used as a divider, or to display passwords and other private information, whereas '^' on the other hand, is mainly only used to indicate exponentiation. Considering this, as expected "log base" has the lowest count, whereas "-", "*", "+", and "over" have the highest counts in the pre-training dataset. Other than performing better on symbols/representations it has seen more often, we also expect that the model will perform better on easier arithmetic calculations, that is calculations involving smaller numbers and operations such as addition and subtraction.

Table 1: Arithmetic operators and their possible representations analyzed

Operator	Representation 1	Representation 2	Representation 3
Addition (+)	+	plus	add
Subtraction (-)	-	minus	subtract
Multiplication (*)	*	times	multiply
Division (\)	\	divided by	over
Exponentiation (^)	^	to the power of	**
Logarithm (log)	log base		



(a) Frequency of occurrence of arithmetic operators, by aggregating all their representations

(b) Frequency of occurrence of all representations of the arithmetic operators

Figure 4: Comparative analysis of frequency of the chosen arithmetic operators and their different representations

4.3 Analysis of Model Accuracy Comparison Across Different Numbers of Digits and Different Representations

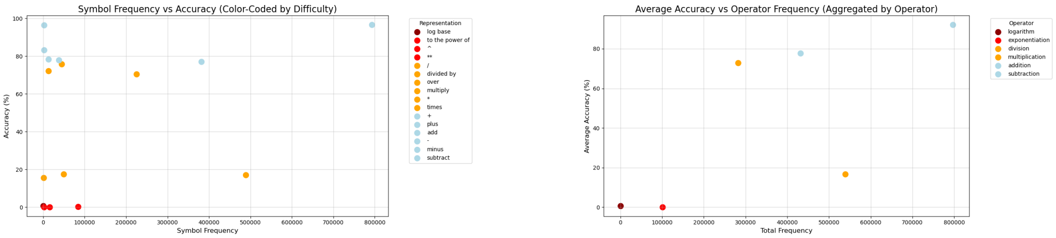
We examined the relationship between the accuracy of model inferences on mathematical problems and three key factors: the number of digits, numerical representation, symbolic representation. Additionally, we explored how accuracy correlates with the frequency of numbers and equations.

4.3.1 Model Accuracy Relative to Operator Representation

Figure 5a plots the performance of the model on all arithmetic questions using a specific operation representation vs. the frequency of that operation representation. Figure 5b similarly plots the performance of the model on all arithmetic questions using a specific operation (average of all its representations) vs. the frequency of that operation in the training data (sum of all its representations).The first plot does not show any indication of a relationship between representation frequency and model performance. We notice that the model achieves a frequency of close to 100% even for representations with very low frequency. However, there does seem to be a relationship between operation type and accuracy. It seems that the easier the operation (here the lighter the dot color on the plot), the higher the accuracy of the model. In both plots we can notice that the model has high accuracy on easier operations (like addition and multiplication) and a generally lower accuracy on harder operations (like log). This corresponds to the fact that the dark red dots are in the lower left of the plots and the light blue and yellow dots are towards the top right of the plot. We assigned difficulty levels based on the chronology of when these operations are taught to humans in school. The difficulty levels are shown in Table 2.

Table 2: Operations, Their Difficulty Levels, and Corresponding Colors

Operator	**Difficulty Level**	**Color**
Logarithm	Most difficult	Dark Red
Exponentiation	Second most difficult	Red
Division	Easier	Orange
Multiplication	Easier	Orange
Addition	Easiest	Light Blue
Subtraction	Easiest	Light Blue



(a) Accuracy vs Frequency of Each Operation Representation Colored by Operation Difficulty Level

(b) Accuracy vs Frequency of Each Operation Colored by Operation Difficulty Level

Figure 5: Accuracy of model on single-digit arithmetic operations vs frequency of operator representation in the pre-training data. Colored based on difficulty level of arithmetic operation

4.3.2 Model accuracy relative to frequency of different digit numbers

As illustrated in Figure 6, there is a strong positive correlation between the model's arithmetic accuracy and the logarithm of a number's occurrence frequency in the pre-training data. Specifically, when accuracy is plotted against the log-transformed frequency of numbers with different digit lengths, a clear linear trend emerges. By applying linear regression, we observe that the model's accuracy improves consistently with increased log-frequency, indicating that repeated exposure to specific numbers during training significantly enhances the model's ability to solve arithmetic tasks involving those numbers.

The relationship between token frequency and accuracy is modeled as follows:

$$\text{Accuracy} = \alpha \cdot \log_{10}(T \cdot F) - \beta, \tag{1}$$

where $\alpha = 0.12$ is the scaling coefficient, $\beta = 0.60$ is the bias term, T represents the total number of tokens in the corpus, and F denotes the frequency of the specific token.

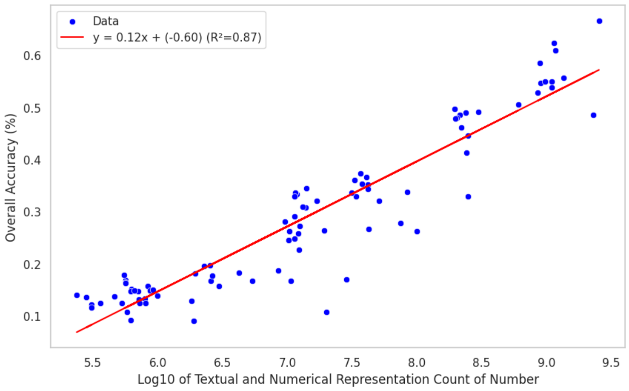


Figure 6: accuracy of calculation for each dataset, showing relation between log of occurrence frequency of numbers in pre-training data and calculation accuracy for both number itself and correspondiong range

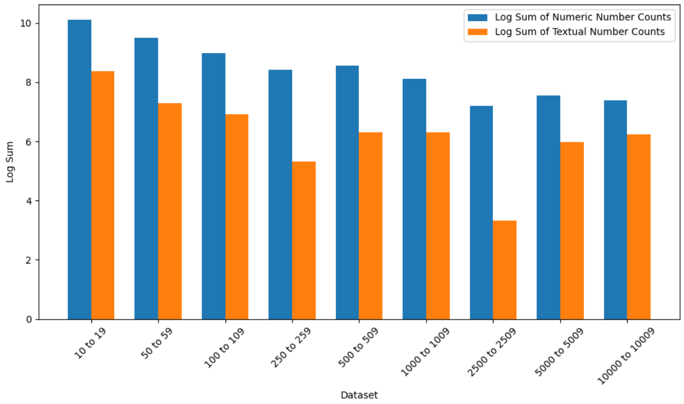


Figure 7: Number of occurrences of numeric and textual numeric representations in the pre-training data for different datasets

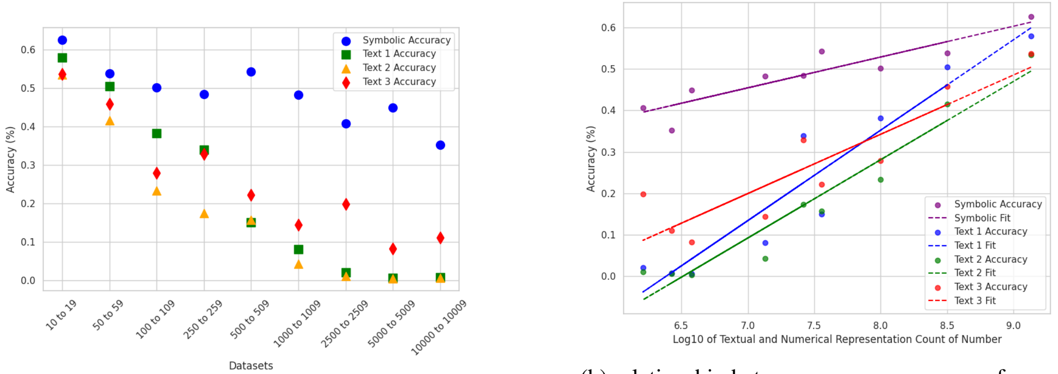
4.3.3 Relationship Between Accuracy and Frequency Across Different Number Representations

Our analysis extends beyond purely numeric forms to include numbers represented partially or fully in text. Across all these representation types—fully symbolic (e.g., "10 + 5"), mixed numeric-text (e.g., "ten + 5" or "10 + five"), and fully textual (e.g., "ten + five").

Frequency Correlation with Accuracy of Different Number Representations: The accuracy of operations with purely numeric, partially textual, and fully textual representations were all positively correlated with frequency. We analyzed the frequency of different numerical representations across various datasets within the pre-training data. As illustrated in Figure 7, numerical representations appear significantly more frequently than their textual counterparts across all datasets. The observed differences range from several hundred to tens of thousands of occurrences, highlighting the dominance of numerical formats in the training corpus. As Figure 8b shows, our results clearly indicate a positive correlation between the frequency of each representation in the pre-training data and the model's arithmetic accuracy.

Symbolic Representations Perform Better: Overall, the accuracy of fully symbolic representation is higher than the accuracy of operations using textual representations of numbers. As Figure 8a demonstrates, fully symbolic representations (e.g., "10 + 5") consistently outperform textual representations (e.g., "ten + five") across all number ranges. This suggests that the model's arithmetic performance is more robust when dealing with familiar, digit-based formats commonly encountered during pre-training. According to Figure 7, this performance difference can be attributed to the higher frequency of numeric forms in the pre-training corpus.

Textual Advantage for Complex Numbers: A fully textual representation of the numbers has a higher performance for larger numbers, with at least more than 3 digits, as well as numbers with fewer zero-digits. Interestingly, as shown in Figure 8a, for certain subsets of larger numbers—particularly those exceeding three digits and containing fewer zeros—fully textual representations demonstrate an accuracy advantage over mixed text-numeric forms. Despite being less frequent in the pre-training data, these textual forms achieve higher accuracy, suggesting that the model may rely on reasoning rather than simple memorization when dealing with more complex numbers.



(a) Average accuracy correspond to different numerical representation on different dataset

(b) relationship between average accuracy of operations contain different representations of numbers and frequency of numbers

Figure 8: The impact of numerical representations and number frequency on average accuracy

4.3.4 Relationship between accuracy and frequency of different operator representation

To determine whether the model's arithmetic performance is driven by mere memorization of familiar operator forms, we examined how accuracy changes with different operator representations. As Figure 9 shows, using the "plus" operation as a case study, we compared symbolic ("+"), textual ("plus"), and alternate textual ("add") variants. Despite these representations appearing at significantly different frequencies in the pre-training data, their associated accuracies did not follow a similar frequency-dependent pattern.

To isolate operator effects from those of number frequency, we focused on arithmetic questions involving numbers with closely matched frequency ranges, such as the sets [251, 252, 253, 254, 257, 258, 259] and [501, 502, 503, 504, 505, 506, 507, 508, 509]. Within these controlled conditions, we found that the model achieved consistently high accuracy across all operator representations, even those that were comparatively rare in the training corpus.

This result suggests that once the model has sufficiently "learned" a particular arithmetic operation, the specific symbol or textual form used to represent that operator no longer critically influences its ability to solve the problem. Instead, the model's arithmetic competence for a given operation seems more resilient and does not degrade simply because the operator is less familiar or less frequently observed. In other words, operator representation does not appear to be the limiting factor for accuracy, which indicates the model may possess a level of abstract reasoning about the operation itself rather than relying solely on memorized patterns of frequently seen operator symbols.

5 Conclusion and Future Work

Our study shows that the frequency of a number in an LLM's pre-training data strongly influences the model's accuracy on arithmetic tasks involving that number. Moreover, how the number is represented—purely numeric, mixed, or fully textual—further modulates performance. In contrast, the choice of operator representation has negligible impact on accuracy. Instead, the intrinsic difficulty of the arithmetic operation itself emerges as the more critical factor shaping the model's success.

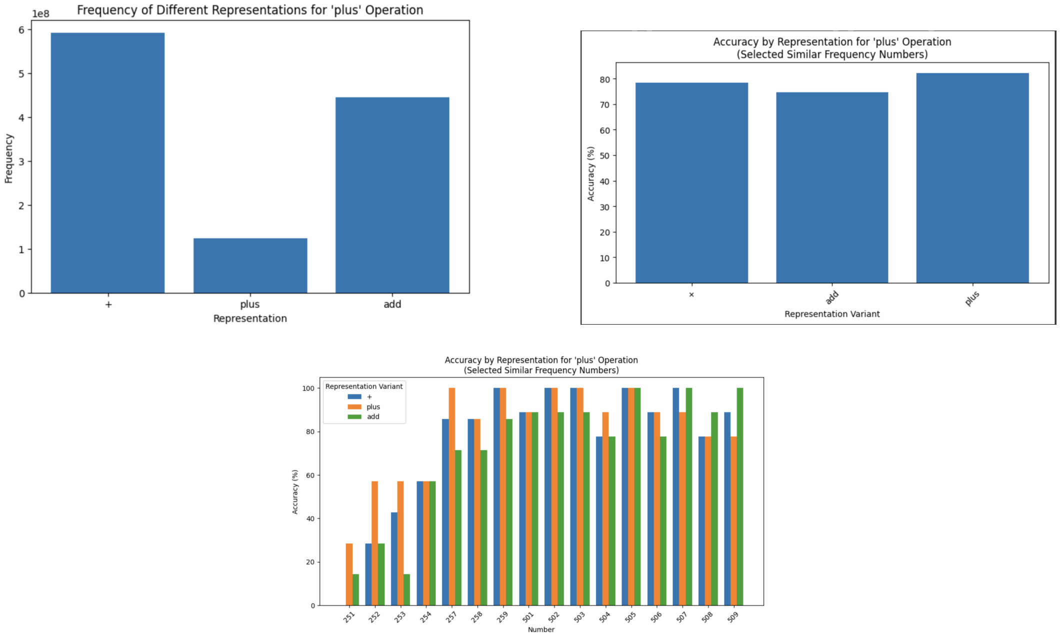


Figure 9: The various frequency of different expression of "addition" and llama's response's accuracy under different expression of addition within set of numbers have close frequency

These findings may suggest that LLMs do not rely solely on memorization of frequently seen patterns; rather, their arithmetic performance may also reflects an underlying capacity for reasoning. In future work, we aim to refine our counting tools to distinguish between mathematical and non-mathematical contexts, enabling more precise frequency measurements. Such improvements would help us better isolate the conditions under which LLMs are truly reasoning, rather than simply retrieving memorized associations. Additionally, we plan to further investigate model's reasoning ability in math by using chain-of-thought prompting which encourage model produce step by step reasoning process before giving its final answer. For example, provide example of worked-out solution that shows intermediate steps and ask model explain reason on new problem. By analyzing correctness and coherence of these intermediate steps, we may further assess whether model is logically deducing results or merely guessing.

6 Challenges and Mitigations

A big limitation of our project is that we are looking at the count of specific words or symbols over the whole pre-training data, regardless of the context. Meaning, even if the word "over" was not used in the context of division, we are still counting that occurrence and using that count metric for our analysis and visualizations. However, we would expect that a model's accuracy in an arithmetic operation would improve if it were to see that symbol more in the context of the arithmetic operation. Infini-gram, which is the tool we are using for retrieving counts, does not allow for us to do filtering based on the context in which a specific symbol or string was used, so this will remain a limitation of our project.

Another challenge or limitation is that there are many more possible representations for the operations we are analyzing. For example division can be expressed using a fraction and exponentiation can be expressed through an exponent (eg 4^7). Some of these notations are harder to query for, which is why we did not use them in our analysis, but figuring out a way to include them would be beneficial to our project.

References

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. A survey on evaluation of large language models. *CoRR*, abs/2307.03109, 2023. URL `https://arxiv.org/pdf/2307.03109`.

Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit cot to implicit cot: Learning to internalize cot step by step, 2024. URL `https://arxiv.org/abs/2405.14838`.

Simon Frieder, Julius Berner, Philipp Petersen, and Thomas Lukasiewicz. Large language models for mathematicians. *Internationale Mathematische Nachrichten*, 254:1–20, 2023. URL `https://arxiv.org/pdf/2312.04556`.

Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens, 2024. URL `https://arxiv.org/abs/2401.17377`.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models, October 2024. URL `https://arxiv.org/pdf/2410.05229`.

Alberto Testolin. Can neural networks do arithmetic? a survey on the elementary numerical skills of state-of-the-art deep learning models. *CoRR*, abs/2303.07735, 2023. URL `https://arxiv.org/pdf/2303.07735`.