A Study of Large Language Models' Ability to Perform Basic Math Operations in Different Bases

Yu-Shu Chen
University of California, San Diego
La Jolla, CA 92093
`yuc067@ucsd.edu`

Abstract

Large Language Models have demonstrated remarkable proficiency in a variety of natural language processing tasks, including translation, question-answering, and fill-in-the-blank exercises, with some evidence of reasoning abilities. In this paper, I explore their mathematical reasoning abilities through their performance on simple mathematical tasks. By evaluating GPT-3.5 turbo and GPT-4o on tasks involving addition, subtraction, and multiplication in uncommon numerical bases, this study aims to shed light on the models' ability to generalize to unfamiliar data and their underlying reasoning potential. The results offer insights into the strengths and limitations of LLMs in performing arithmetic operations outside the familiar base-10 system, contributing to a deeper understanding of their reasoning capacities.

1 Introduction

Large Language models (LLMs) have become exceptionally powerful in their ability to understand and generate human-like text. Advanced models, such as the Generative Pre-trained Transformer (GPT), can proficiently handle a wide array of NLP tasks—including translation, question-answering, and fill-in-the-blank tasks—while also exhibiting some capacity for reasoning. These models are expanding with increasing data and parameters, pushing the boundaries of what they can achieve. However, adapting these models to new information or domains often requires retraining or fine-tuning, processes that are both resource-intensive and time-consuming. Despite utilizing millions or even billions of parameters, their cognitive capacity still remains limited compared to the human brain, from which they were inspired.

As we aim to move towards Artificial General Intelligence (AGI), one critical aspect is the model's ability to apply its existing knowledge to new, related tasks. In this study, I investigate how well models perform arithmetic tasks, particularly focusing on basic arithmetic calculations in different numerical bases. This task is chosen because I anticipate that the model has not been extensively trained on such tasks. However, the models are trained to perform basic arithmetic calculations in base 10 and possess knowledge of base conversions and how different bases operate. Therefore, if the models are able to demonstrate reasoning capabilities, it should be able to perform these tasks decently well.

2 Dataset

The dataset consists of 840,000 problems covering integer addition, subtraction, multiplication, and division. The dataset was created by randomly generating 6,000 pairs of integer numbers and using them to create problems for each arithmetic operation. These problems were then converted into numerical bases ranging from 2 to 36, covering all possible bases represented by numbers and letters in the alphabet. The integers generated range from 0 to 46,655 (36^3-1), which is intended to test the model's ability to handle carry-overs across different bases.

The problems are formatted as follows:

```
(number_1)_{(base_1)}(operator)(number_2)_{(base_2)}=
```

, where: `(number_1)` and `(number_2)` are the integer values being operated on, `(base_1)` and `(base_2)` indicate the numerical bases of each number, and `(operator)` represents the arithmetic operation (e.g., +, -, *, /).

For example, a problem would look like this:

```
110001010011111_{2}+110101110101010_{2}=
```

3 Methods

For this study, I chose OpenAI's GPT-3.5 Turbo and GPT-4o to test, as they are among the best-performing LLMs currently available. Since the task involved only simple arithmetic operations, I did not pretrain the models. The following message was passed to the models' APIs for each equation:

```
equation = "110001010011111_{2}+110101110101010_{2}=" # for example
messages=[
    {"role": "system",
     "content": "You will be provided with an equation, and your task is to
         compute its correct answer. At the last line of your response,
         summarize your response by showing the original equation, the
         intermediate steps, and the final result, in one line in text with
         each number followed by its base in the format number_{base}."},
    {"role": "user", "content": equation}]
```

Due to resource constraints, including both labeling and API costs, I sent 10 problems per base (ranging from 2 to 16) for arithmetic operations of addition, subtraction, and multiplication to both models, totaling 900 data points. Division was excluded because it results in decimal numbers, which complicates the evaluation process for the bases. This amount of data was deemed sufficient for testing the modes' ability to perform these operations effectively, similar to how one might assess a person's proficiency in these arithmetic tasks.

To label the responses, I reviewed each response from the models and documented the most accurate and generous interpretation of its process and results. This process was conducted twice: first by extracting the summary line provided by the model as requested in the prompt, and second, by evaluating the entire response for consistency and accuracy.

4 Analysis

In this section, I present a comprehensive analysis of the responses generated by GPT-3.5 Turbo and GPT-4o to understand what the models are good at and where they may have limitations.

4.1 Correctness & correctness with process

This subsection evaluates two related aspects of the models' performance: the correctness of the answers and the correctness in relation to the provided process. I assess whether the responses were correct or incorrect and how well the models followed the steps required for solving the problems. Points are awarded based on the accuracy of the provided process, and we examine how this impacts the overall correctness of the responses.

The charts suggest that GPT-4o has strong performance in base-10 addition, but its performance significantly drops in other bases. This indicates that the model's reasoning abilities are heavily biased towards familiar numerical representations, highlighting a potential area for improvement.

The model shows almost no correct answers across all bases for multiplication, with a slight exception for base 10, where it achieves only 10% accuracy. This suggests that the model struggles significantly

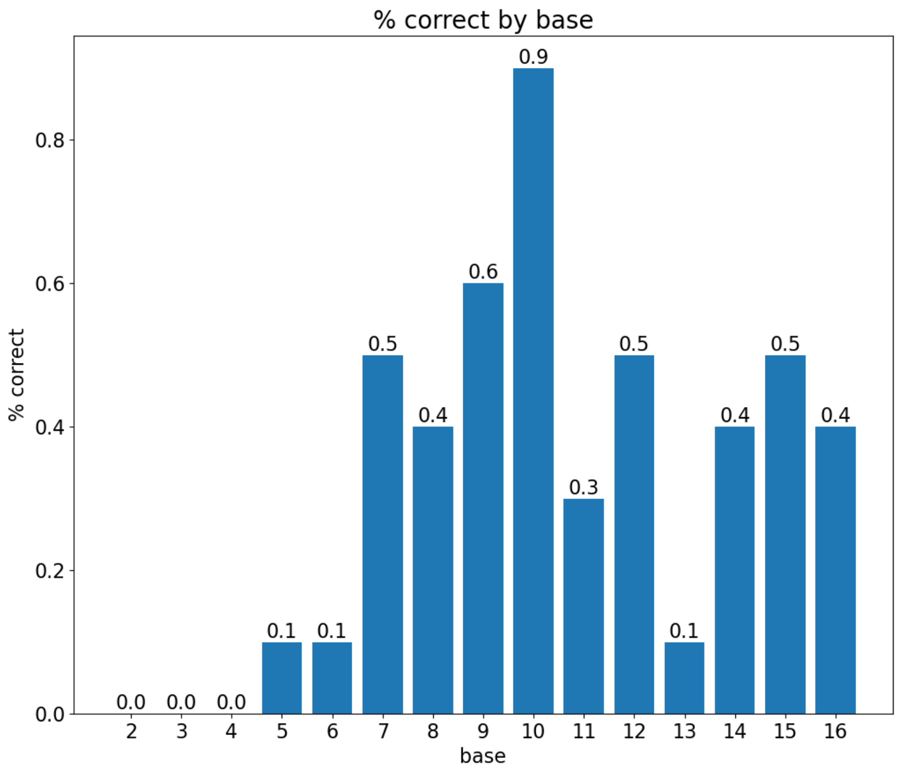

Figure 1: addition

with multiplication tasks, especially when the operations are in non-decimal bases. The scores weighted by process are uniformly low across all bases, with the highest scores barely reaching 0.15 in bases 9, 11, and 14. Even in base 10, where the model shows a small uptick in correctness, the score remains low at 0.12, indicating a very weak performance.

4.2 Numerical correctness

This subsection evaluates the difference between the correct answers and the answers from the responses.

Addition The box plots for addition illustrate the difference between the actual and predicted results for addition across various bases. The x-axis represents the difference between the actual result and the model's prediction, while each subplot corresponds to a different base.

The 'all' plot at the top provides a consolidated view, showing that the differences mostly lie within a range of ±20, though there are more extreme values as well. For the other bases:

Base 10: The plot for base 10 shows a very narrow range of differences around zero, indicating that the model performs well in base 10 addition with minimal deviation.

Bases 12 & 14: For bases 12 and 14, the differences remain small, with predictions closely aligning with the correct answers. This suggests that the model generalizes slightly better in these bases compared to others.

Bases 3, 6, 8, 9, 13, and 16: The differences in these bases show a broader range, but without the extreme outliers. Bases like 3 and 8 still exhibit larger ranges in prediction errors, indicating that the model struggles with these bases more than others.

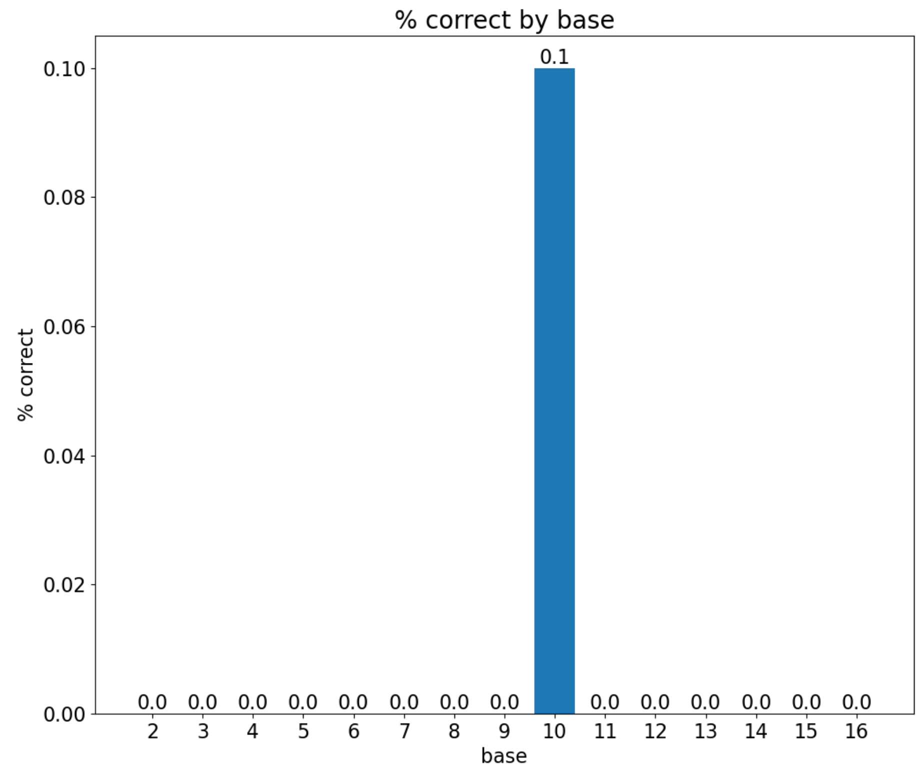

Figure 2: multiplication

Base 7: Base 7 has a very narrow spread but is consistently negative, indicating that while the model is consistently off, it underpredicts by a small amount.

Multiplication The box plots for multiplication illustrate the difference between the actual and predicted results for addition across various bases. The x-axis represents the difference between the actual result and the model's prediction, while each subplot corresponds to a different base. It can be seen here that a lot of the difference for multiplication are orders of magnitudes off from the correct answers.

4.3 Character-wise correctness

This subsection assesses the model's performance by the percentage of character differences between the predicted and correct results. This analysis is important because it helps us understand how discrepancies in predictions can vary depending on the magnitude of the numbers involved. Mistakes made in calculations involving larger numbers should not be disproportionately penalized compared to errors in smaller numbers. This approach allows for a more nuanced understanding of model performance and aids in identifying specific areas where improvements are needed.

It can be seen here that a much higher percentage of characters are correct than the proportion of correct results when evaluating the overall correctness. This discrepancy highlights several key points:

- The model is capable of generating accurate components of the solution but struggles with aggregating these components into a fully correct answer.

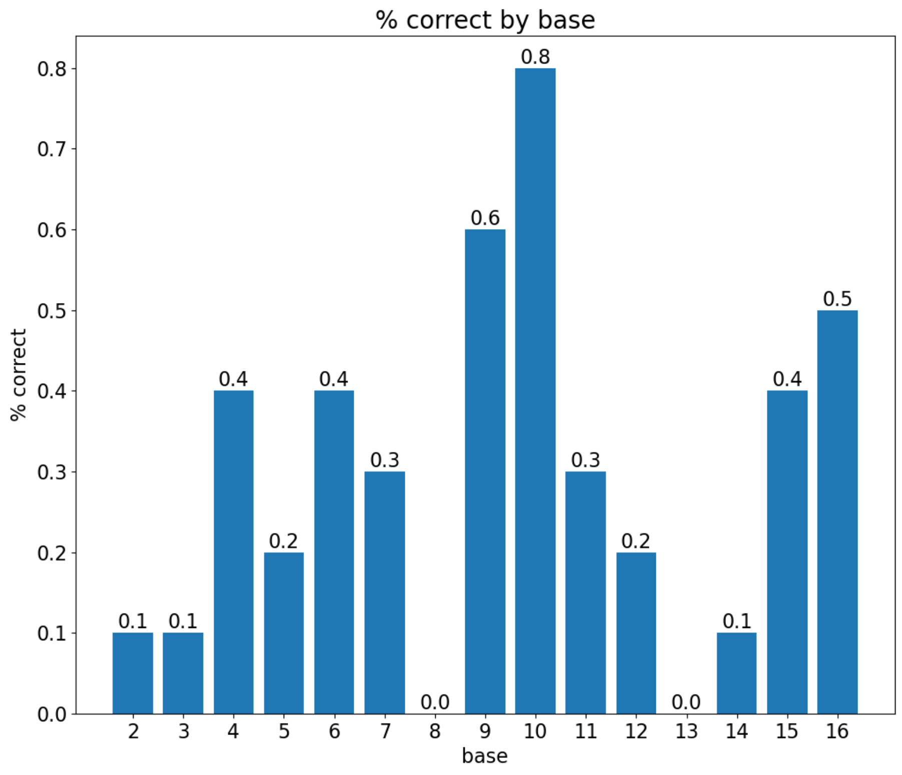

Figure 3: subtraction

- A small mistake in one stage can lead to a larger discrepancy in the final result. Even if a significant portion of characters is correct, the cumulative effect of errors at various stages can result in an incorrect overall result.
- This disparity underscores the importance of evaluating models not just on the correctness of final results but also on the accuracy of intermediate stages and individual components. In scenarios where partial correctness is high but overall correctness is low, it indicates that improvements are needed in how the model integrates individual characters into complete solutions.
- To address this issue, enhancing the model's ability to handle arithmetic operations more accurately across different bases and ensuring better integration of correct characters into the final answer are essential. This might involve refining the model's internal algorithms for arithmetic operations or increasing its robustness to handle errors in intermediate calculations more effectively.

In the case of multiplication tasks, the graph reveals a lower percentage of correct characters compared to the overall correctness. Despite this lower percentage, the same underlying concept applies.

4.4 steps analysis

- something and something else

4.5 Errors and their Categorization

- something and something else

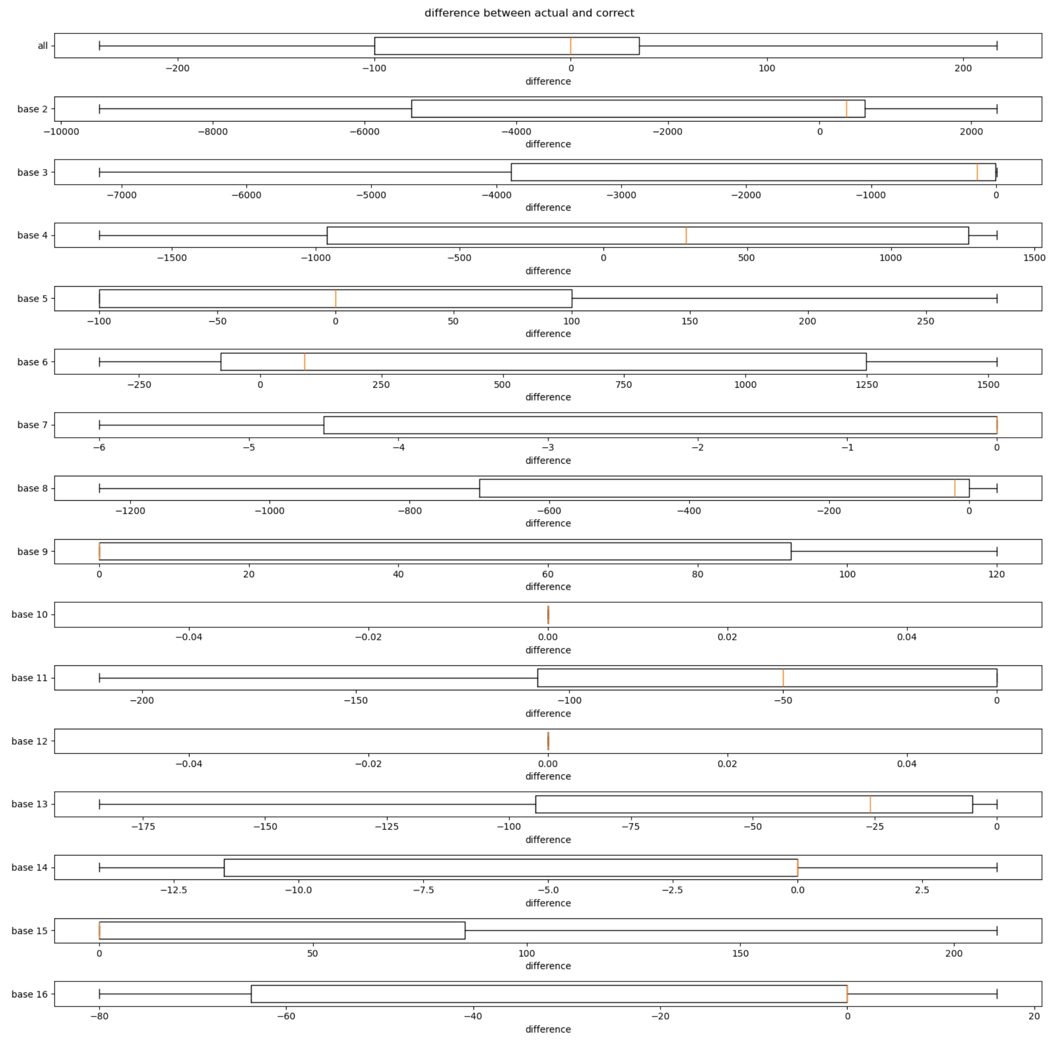

Figure 4: addition

5 Conclusion

In conclusion, the analysis conducted above demonstrates that current LLMs lack the capability to accurately perform arithmetic operations in bases other than 10, with multiplication being particularly challenging. This limitation highlights the need for further research and development to enhance LLMs' ability to generalize across different numerical bases. Future work could focus on improving model architectures, incorporating specialized training data, or developing novel algorithms that enable more accurate and consistent performance in non-decimal bases. By addressing these gaps, LLMs could become more versatile tools for a wider range of mathematical applications.

References

[1] Brown, Tom B., et al. "Language Models Are Few-Shot Learners." arXiv.Org, 22 July 2020, arxiv.org/abs/2005.14165.

[2] Khosla, Savya, et al. "Survey on Memory-Augmented Neural Networks: Cognitive Insights to Ai Applications." arXiv.Org, 13 Dec. 2023, arxiv.org/abs/2312.06141.

[3] Schick, Timo, et al. "Toolformer: Language Models Can Teach Themselves to Use Tools." arXiv.Org, 9 Feb. 2023, arxiv.org/abs/2302.04761.

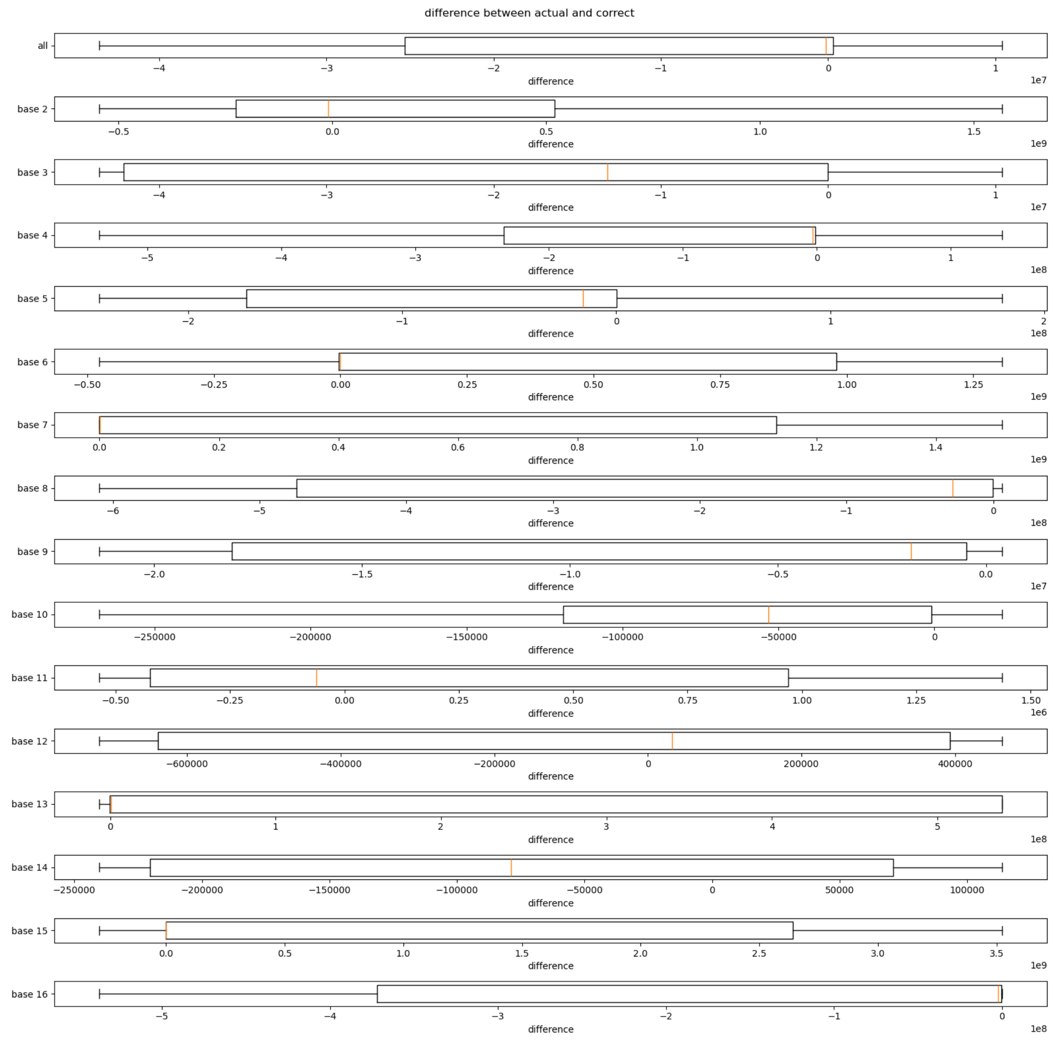

Figure 5: multiplication

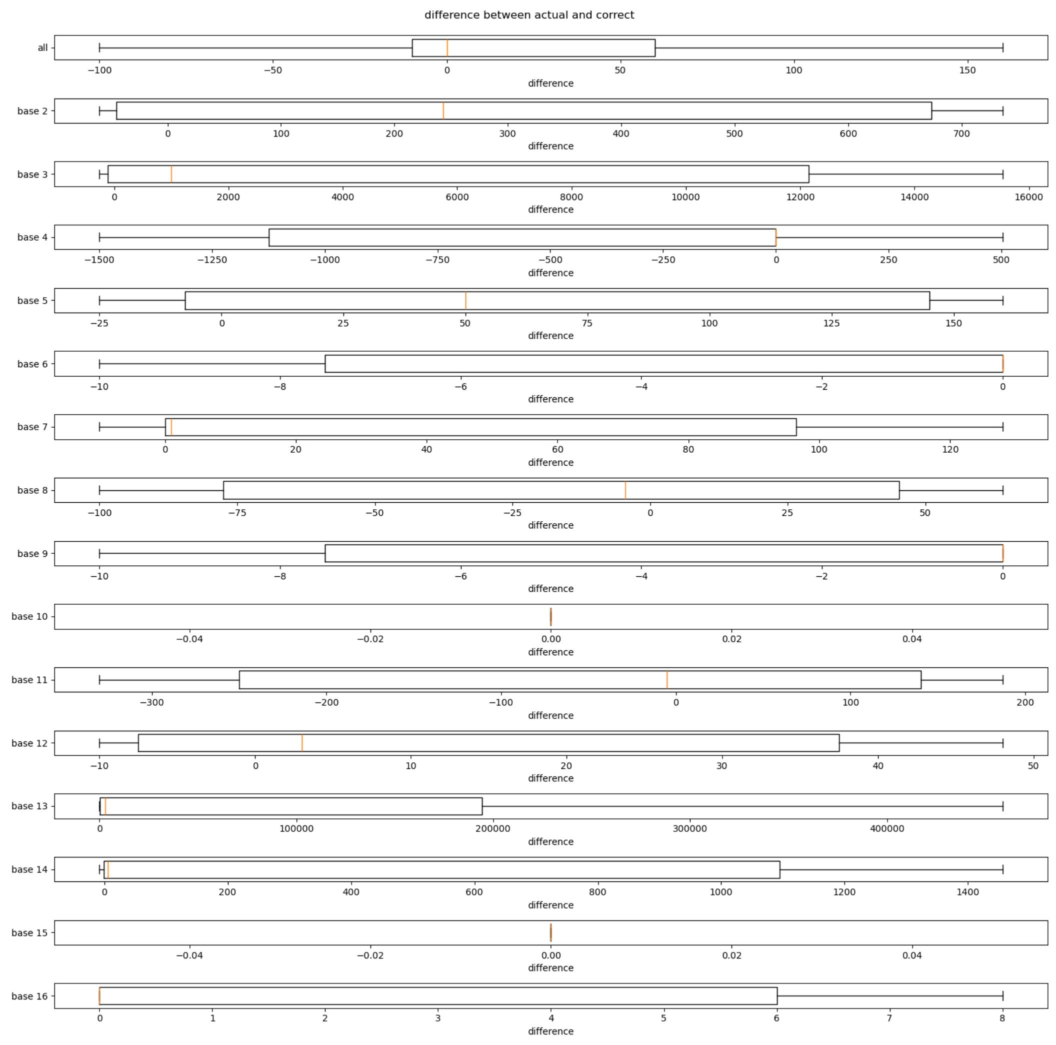

Figure 6: subtraction

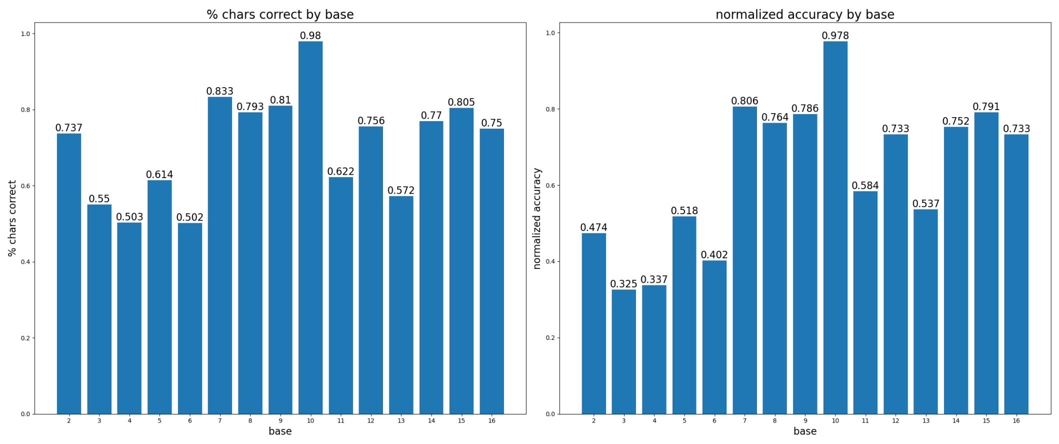

Figure 7: addition

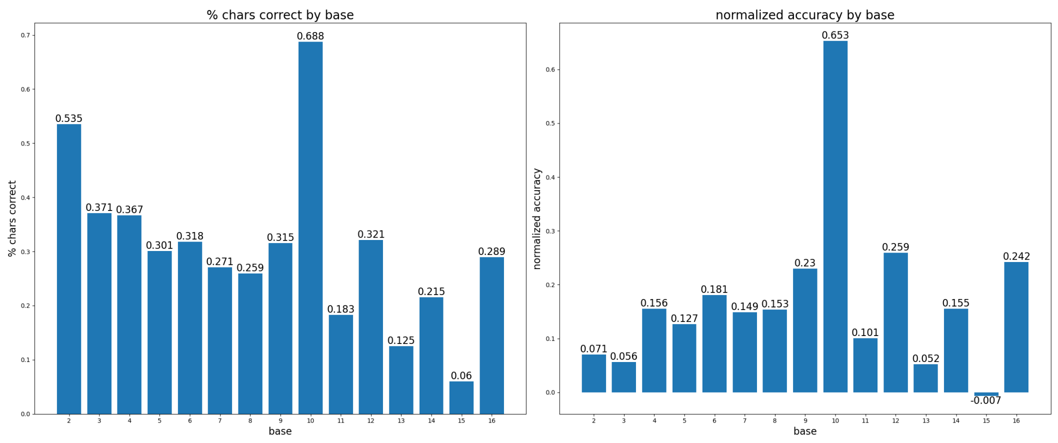

Figure 8: multiplication

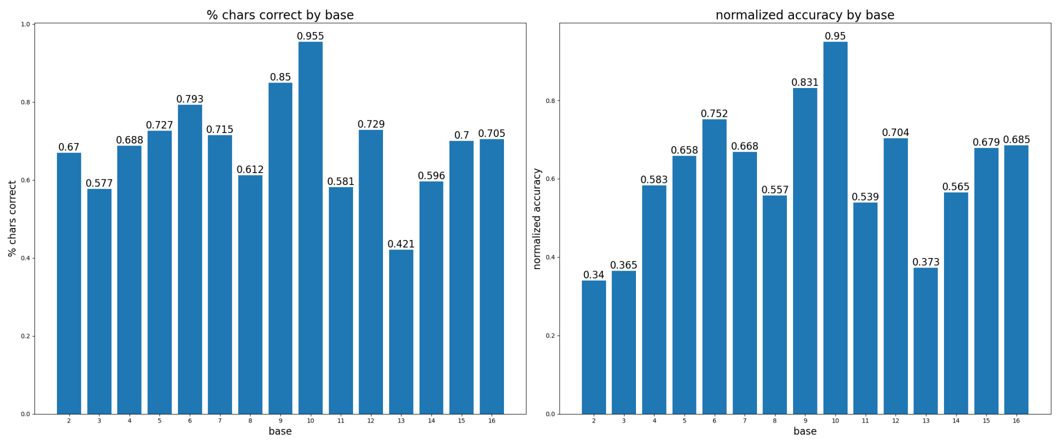

Figure 9: subtraction