Fourier Embeddings and Their Role In LLM Arithmetic

Daniel Warren A18138846
Anthony Tong A17720195
Gavin Simmons A17093287
Chi Zhang A16346955

Christopher Rebollar-Ramirez A16982224

Abstract

This research project aims to reproduce and expand upon the findings of Zhou et al. (2024, 2025) regarding how large language models (LLMs) perform arithmetic operations. Recent research suggests that LLMs leverage Fourier features rather than standard Byte-Pair encoding tokenization to perform arithmetic, with multi-layer perceptrons (MLPs) identifying broad numerical structures and attention layers capturing modular properties, which we seek to validate on the GPT-2 architecture, a model structurally distinct from the Llama-3.2 based models used in the papers. Additionally, we will implement and evaluate the FoNE architecture as described in Zhou et al (2025), which introduces Fourier-based number embeddings to improve arithmetic computation, both in accuracy and efficiency. Our plan involves generating a custom dataset of arithmetic problems, then starting with less complex operands and progressively increasing complexity. Resources permitting, we will continue to explore broader implications for general arithmetic operations in LLMs. This work contributes to the growing collection of research seeking to understand the learning and reasoning processes of LLMs through the lens of quantifiable arithmetic capabilities.

1 Introduction

Recent large-scale training of large language models (LLMs) has led to impressive capabilities across a variety of downstream tasks, yet the fundamental mechanisms behind learning have yet to be well understood. In particular, pre-trained LLMs have an emergent capability to perform arithmetic operations, with varying degrees of success. Numerous approaches have been proposed to address the task of basic arithmetic in LLMs, but most either ignore the models' inherent capabilities in performing computation or directly engineer training methods that are rather artificial and hacky, and often based on empirical performance [1][3][2]. However, recent work by Zhou et al. proposes that the natural emergence of Fourier features in the embeddings and activation logits of pre-trained LLMs is involved in its inherent computational power [5]. A followup work demonstrates the ease of training when initializing Fourier embeddings in models for performing basic arithmetic operations [6].

While these studies demonstrated the effectiveness of the Fourier Number Embedding (FoNE) scheme in the LLaMA-3.2 model, it remains unclear whether the same benefits hold for other widely used architectures such as GPT-2. In this project, we aim to reproduce and extend these findings by assessing whether FoNE also improved numerical computation efficiency when applied on GPT-2. In this report, we show that GPT-2 small [4], a 124M-parameter tiny LLM, is capable of arithmetic operations generalizable to large numbers through a modification of numerical processing inspired from [5]. In particular, we investigated the effectiveness of division in LLMs through generating a

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

new dataset focused on division between pairs of numbers and trained our FoNE-based model based on that. Our experiments explored whether GPT-2's inherent numerical representations align with prior work and investigated the feasibility of using FoNE for more complex computations beyond addition, such as integer division.

2 Related Work

Recent work has shown that the traditional next-token prediction learning algorithms are ineffective in training LLMs to perform arithmetic; instead, many models used alternative approaches to often learn the computation methods and rules internalized in transformer weights. Deng et al. [1] tackled 9x9 digit multiplication in GPT-2 Small by feeding chain-of-thought (CoT) prompting, and gradually removing tokens until the model learned to internalize the process. Lee et al. compared the addition task to learning a rank-2 matrix completion problem, where an $n \times n$ matrix was filled in with the training data of $i + j$ corresponding only to the i-th row and j-th column [3]. Lee also noticed that the other arithmetic operations were not necessarily the same – in particular, subtraction was a substantially different task, since the order of operations was no longer commutative [3]. Lee also reinforced the importance of CoT sampling and the power of intermediate steps [3]. Another widely used approach employs a tool-integrated reasoning agent (ToRA) to relieve the model of computational tasks, instead burdening the model with code generation which can then be run in the background [2]. While an effective method for obtaining computer-level precision, the complexity of implementation and the reliance on calculators and symbolic solvers suggest that there are likely simpler methods to recover the solutions internally.

In general, conventional architectures and training methods struggle with arithmetic, particularly when numbers extend beyond their training set. Traditionally, LLMs tokenize numbers using subword or digit-wise tokenization, which often results in fragmented representations for numerical data, and is often not interpretable. In particular, subword tokenization generally arises from byte-pair encoding (BPE) schemes, in which the merge order can often be random and at the mercy of the tokenizer pretraining corpus's number frequencies. Digit-wise tokenization may offer an improvement in stability and accuracy, considering that basic arithmetic operations can be performed step by step with pairs of digits. However, due to the quadratic nature of the attention mechanism, digit-wise tokenization is significantly slower than subword tokenization. Recent work has proposed that LLMs leverage Fourier-like transformations in their attention and MLP layers to better process numerical relationships; additionally, Fourier features in number embeddings emerge from the pretraining of LLMs [5]. Building on this idea, Zhou et al. introduced Fourier Number Embedding (FoNE), which encodes numbers using sinusoidal embeddings before passing them into a LLaMA-3.2 model. This method achieved an accuracy rate of over 99% for addition, subtraction, and multiplication using a relatively small set of training data [6]. The proposed method offers significant improvements over both subword and digit-wise tokenization, incorporating impressive high precision representations of numbers without sacrificing low sequence length.

3 Methods

3.1 Fourier Analysis

For our Fourier analysis methodology, we investigated how GPT-2 processes arithmetic operations by examining the intermediate representations in different layers. We implemented a framework to capture outputs from both MLP and attention modules for specific addition examples (e.g., 154 + 97). These outputs were converted to logit values by multiplying with the model's output embedding matrix, allowing us to visualize activation patterns across layers using heatmaps similar to those in [5]. We then transformed these logits into Fourier space to identify frequency components associated with different numerical operations. By examining the resulting spectral patterns, we identified distinct roles: low-frequency components (periods > 50) for magnitude approximation and high-frequency components (periods ≤ 50) for modular operations. We validated these findings through ablation studies where specific frequency bands were selectively filtered to observe their impact on arithmetic prediction accuracy.

3.2 FoNE

We validate the claims of Zhou et al. (2025) by testing 6 digit integer addition on the FoNE-equipped GPT-2 small model. We remain consistent with [5], using a cosine annealing learning rate scheduler with warmup and max learning rate 0.005, but kept the straightforward integer addition dataset which presents entries in the following format:

$$n_1 + n_2 = [\text{answer}].$$

The dataset is publicly available on HuggingFace at the path `Onlydrinkwater/int_addition2`. We also remain consistent with the addition methods for the multiplication set, and focus on 4 digit multiplication using the HuggingFace dataset `Onlydrinkwater/int_multiplication2` with the following format:

$$n_1 * n_2 = [\text{answer}].$$

We run two primary experiments for the division task on GPT-2. First, we test the feasibility of FoNE on floating point division, allowing up to 5 digits on either side of the decimal point for all values, and restrict $n_1 \in [1, 99999], n_2 \in [1, 9999]$. We then construct the division dataset by ensuring all entries are of the form

$$n_1/n_2 = [\text{answer}].$$

We then simplify to the integer division task, restricting n_1 to any number up to 8 digits and n_2 to 4 digits, where n_1/n_2 is guaranteed to be an integer. We do not consider the negative number case, and the format is identical to the one above. For both synthetic datasets, we follow the dataset structure from earlier experiments, using 720,000 training samples, 80,000 validation samples, and 200,000 test samples. For all experiments, we maintain the learning rate to be up to 0.005 identically to [6], with warmup epochs and a cosine annealing learning rate scheduler. We set batch size to 512 and initialize from pretrained weights, and trained until we ran out of time. We use the publicly available repository linked in the Acknowledgements section for training FoNE.

4 Results

4.1 Emergence of Fourier Embeddings in Pre-Trained Models

The analysis of number embeddings in Fourier space provides strong evidence that pre-trained GPT-2 models inherently develop structured numerical representations. Spectral analysis of these embeddings reveals distinct frequency peaks, suggesting that the model learns to encode numerical relationships in a Fourier-like manner during large-scale pretraining. In contrast, when GPT-2 was trained from scratch on arithmetic tasks, such structured frequency components were not observed, indicating that these embeddings do not emerge solely from task-specific training. This result aligns with previous findings that suggest large language models acquire latent numerical structures during pretraining, which subsequently influence their arithmetic capabilities. The presence of such Fourier features informed Zhou's decision to integrate Fourier Number Embeddings (FoNE), which we show in subsequent experiments. The features in the number embeddings are visualized in Figure 1

4.2 Attention and MLP Logits

Our Fourier analysis reveals distinct computational roles for attention and MLP modules during arithmetic operations. The number embeddings exhibit a Fourier structure with prominent components at periods $T = 520, T = 260, T = 173.33, T = 10, T = 5$, and $T = 2.50$, confirming that numerical representations are inherently encoded with frequency components during pre-training.

The logit heatmaps demonstrate complementary activation patterns across layers. MLP logits show more uniform activations with gradual transitions around the correct answer ($y = 251$), while attention layers exhibit pronounced periodic structures. This supports the mechanistic hypothesis that these components serve distinct computational roles. The heatmaps are given in Figure 2.

When transformed to Fourier space, MLP outputs demonstrate dominant low-frequency components with periods $T = 520$, $T = 260$, and $T = 173.33$, alongside weaker high-frequency signals. These low-frequency components enable MLPs to approximate the magnitude of arithmetic results. The consistent pattern across layers 21-33 indicates that this computational mechanism persists throughout the network depth.

Table 1: Best validation accuracies for vanilla and FoNE training schemes on the integer division task.

Method	Accuracy	Digit-wise Accuracy
Vanilla	5.197%	78.90%
FoNE	75.92%	93.62%

Attention outputs in Fourier space show a similar spectral profile but with distinct differences in specific frequency bands. While also leveraging low-frequency components, attention layers exhibit relatively stronger high-frequency activity that corresponds to modular operations. This aligns with [5], finding that attention layers primarily perform modular arithmetic operations like determining parity (mod 2) or units-place digits (mod 10).

The periodic activation patterns observed in both modules confirm that the model leverages Fourier features for arithmetic computations, with MLPs focusing on magnitude approximation and attention-refining predictions through modular arithmetic. This division of computational labor enables effective mathematical processing across multiple transformer layers.

4.3 FoNE on Addition

The application of FoNE to arithmetic tasks demonstrated its effectiveness in improving numerical computation within GPT-2. In the six-digit addition task, the model achieved 100% accuracy on both training and validation datasets within 33 epochs, with over 99% accuracy reached after only a single epoch. We show the training and validation loss curves as well as the accuracies in Figure 3.

4.4 FoNE on Multiplication

In the four-digit multiplication task, the model approached approximately 62% accuracy after 75 epochs, with a digit-wise accuracy of 95%. We show the training and validation loss curves as well as the accuracies in Figure 4

4.5 FoNE on Division

Due to project time limits, we performed only an initial test run on the decimal division task, and the model reached 0.56% accuracy after 50 epochs with a digit-wise accuracy of 63.80%, without fully converging. On the other hand, the integer division task results are reported; in Table 1, we report the best validation accuracies for the two methods, regular finetuning and FoNE. In Figure 5a we plot both accuracy measures over training epochs, and we report the train and validation losses for both methods in Figure 5b. We recognize that none of these models were able to converge in time.

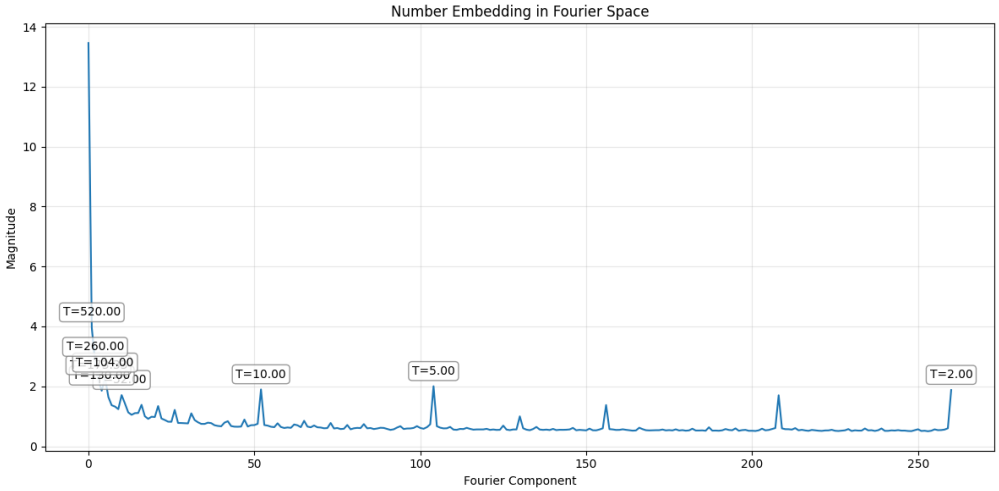
5 Discussion

5.1 Fourier Analysis

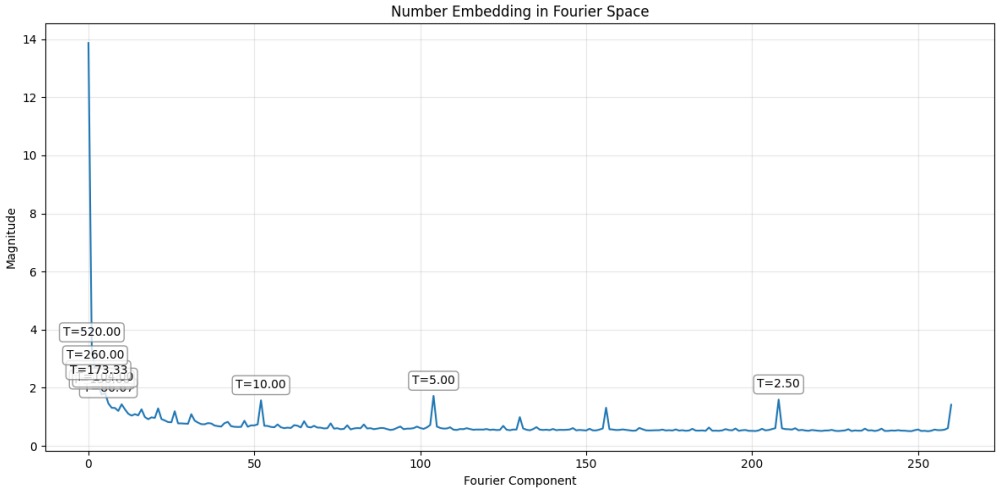
The results from our Fourier Analysis were somewhat consistent with what we anticipated, particularly in terms of observing notable peaks in magnitude of number embedding in Fourier space in the pretrained and pretrained plus finetuned models, but not the model trained from scratch. This aligns with the findings of Zhou et al (2024), suggesting that the presence of these peaks is likely influenced by the model's training history, the Fourier representations were likely learned during pretraining.

In contrast, the model trained from scratch lacks any well defined peaks, likely due to a lack of prior knowledge or structure that is provided by pretraining, which causes an unstructured embedding pattern in Fourier space.

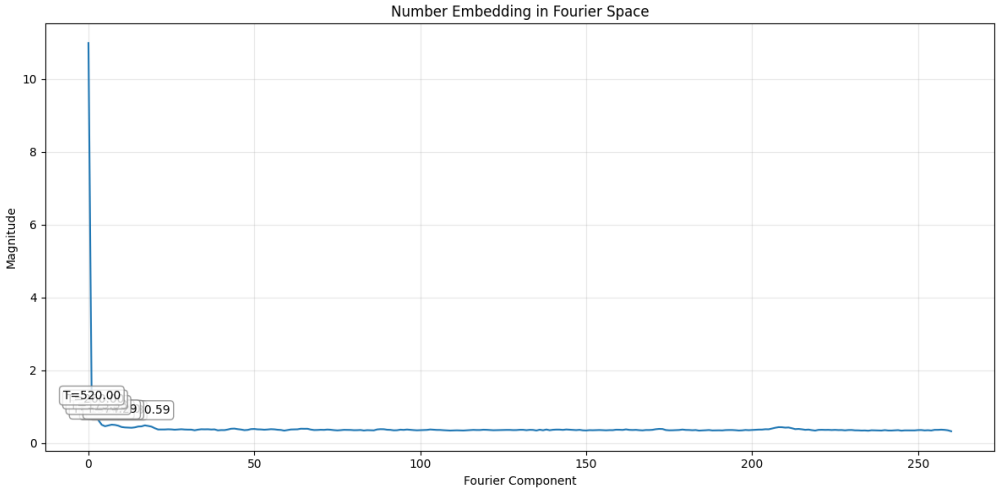
However, while the embeddings in Fourier space showed great promise and accurately reflected the findings in the paper, the heatmaps containing the number of logits across MLP and Attention layers left something to be desired. Although increasing the size of the model, from GPT2-Base to GPT2-Large did move the heatmaps in the correct direction, there are still discrepancies between



(a) Token embeddings in the frequency dimension for the pretrained GPT-2 large weights.



(b) Token embeddings in the frequency dimension for the finetuned from pretrained GPT-2 large weights.



(c) Token embeddings in the frequency dimension for the GPT-2 large model trained from scratch.

Figure 1: Token embeddings visualized in the Fourier space for various GPT-2 large models.

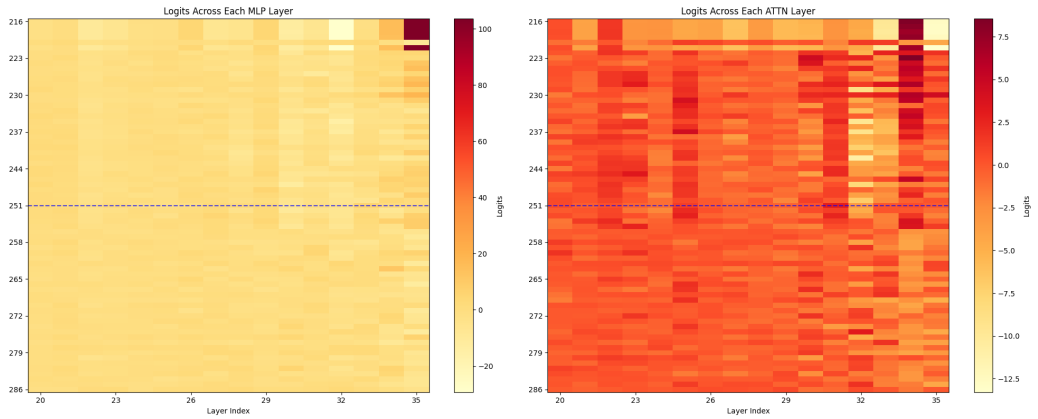


Figure 2: Intermediate MLP and attention logits for the finetuned GPT-2 large model.

our results and the results we were trying to replicate. The most immediate difference between us and the paper is the model used, we used GPT2-Large and the paper used GPT2-XL. We saw an improvement in results when we increased the size from base to large, so presumably increasing the model size more should result in better results.

It is also possible that these results are influenced by other factors, and further investigation would be necessary to identify the root cause of the discrepancies, whether they come from differences in implementation, hyperparameter settings, or architectural choices. A deeper exploration of these aspects would not only provide a clearer understanding of the model's behavior but also allow us to generalize this understanding to other areas, enhancing the overall robustness of our findings.

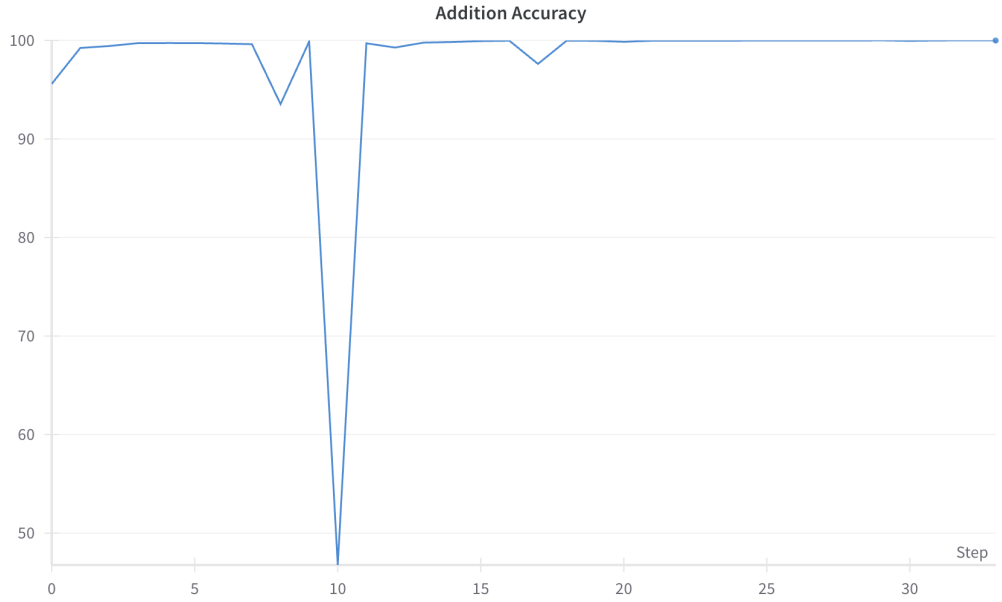
5.2 FoNE Effectiveness

Our results demonstrate that the application of FoNE on GPT-2 small yields highly impressive performance on addition tasks, aligning with Zhou et al. (2025) on the effectiveness of Fourier feature initializations for number embeddings. Indeed, the validation accuracy reached >99% in just one epoch of training, and hit 100% shortly later at epoch 33.

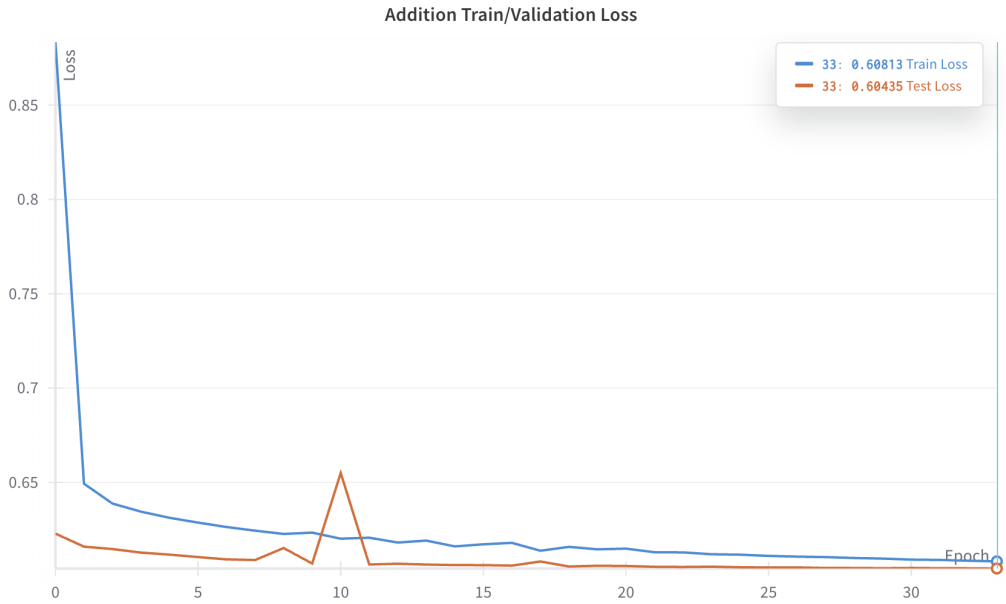
However, performance on multiplication and division did not meet our initial expectations. While the model still showed strong performance in these operations, especially for a model the size of GPT-2 small, the accuracy was significantly lower compared to addition. Certainly, the task of learning multiplication and division is significantly more challenging compared to addition. It should be noted that with the lack of resources it is possible that GPT-2 small could indeed learn these operations to a high accuracy and precision given the proper hyperparameters, but due to project limitations this could not be accomplished. One should note that in particular, while training on the division task, we saw that as the model moved through its warmup epochs closer to the max learning rate region, the performance began to fluctuate; on its first cosine decay, the model began to regain its performance. As such, we suggest that the failure of convergence to 100% should not rest within the FoNE architecture itself, but rather the instability of training with suboptimal hyperparameters.

In particular, we also found significant variance between the performance of decimal and integer division. We propose that, in addition to suboptimal hyperparameters, that in particular non-integer division suffers from the additional truncation/rounding task which it now must adapt to. In particular, simply multiplying numbers by powers of 10 should not impact model performance by the nature of the FoNE architecture, but performing divisions such as $1/3$ can result in significant noise./

Despite these challenges, of course, demonstrating an arithmetic capability in GPT-2 small without significant fine-tuning or special tools used in other papers lends way to insight on the inner workings of LLMs and their learning capabilities. We recognize the significant improvements on the particularly difficult task of division on a limited small GPT-2 model.

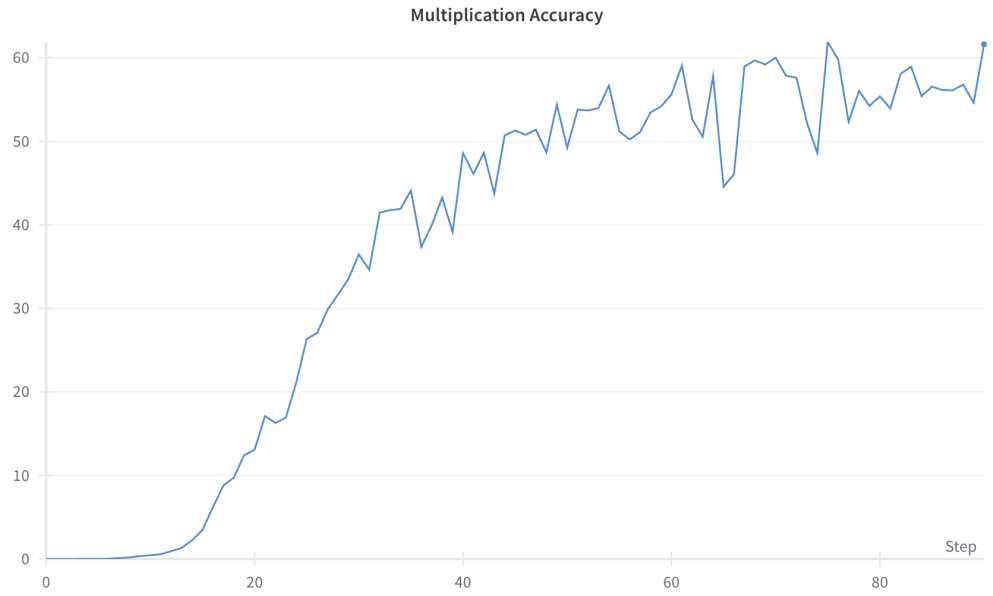


(a) Number and digit accuracies on the addition task for FoNE.

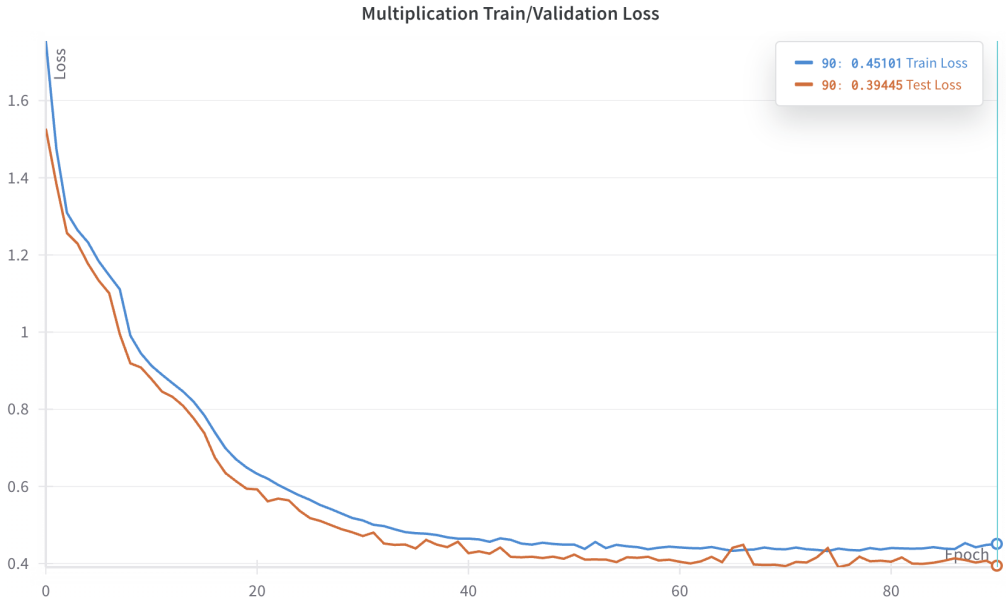


(b) Training and validation losses on the addition task for FoNE.

Figure 3: Performance on the addition task with FoNE.

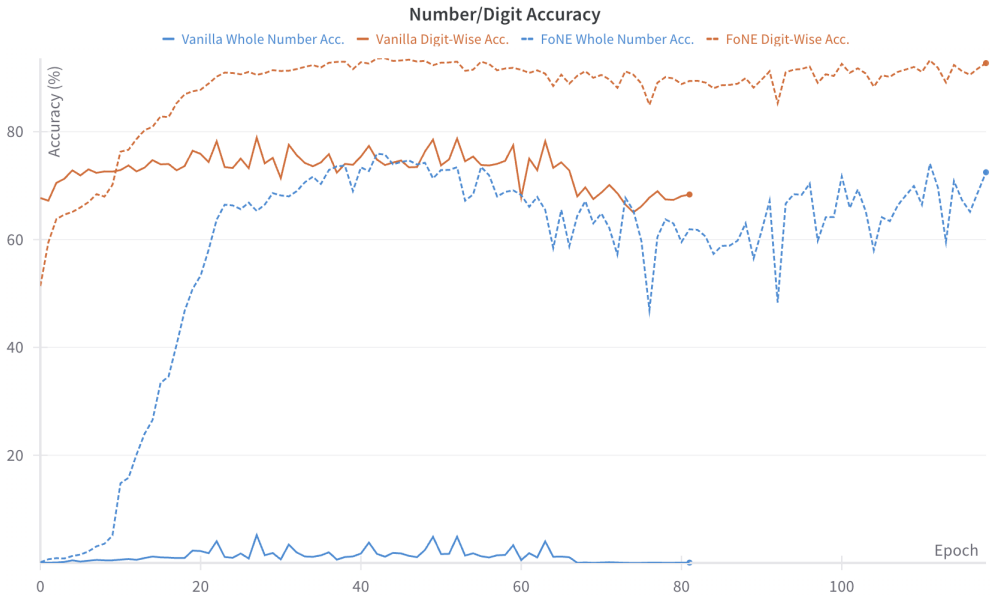


(a) Number and digit accuracies on the multiplication task for FoNE.

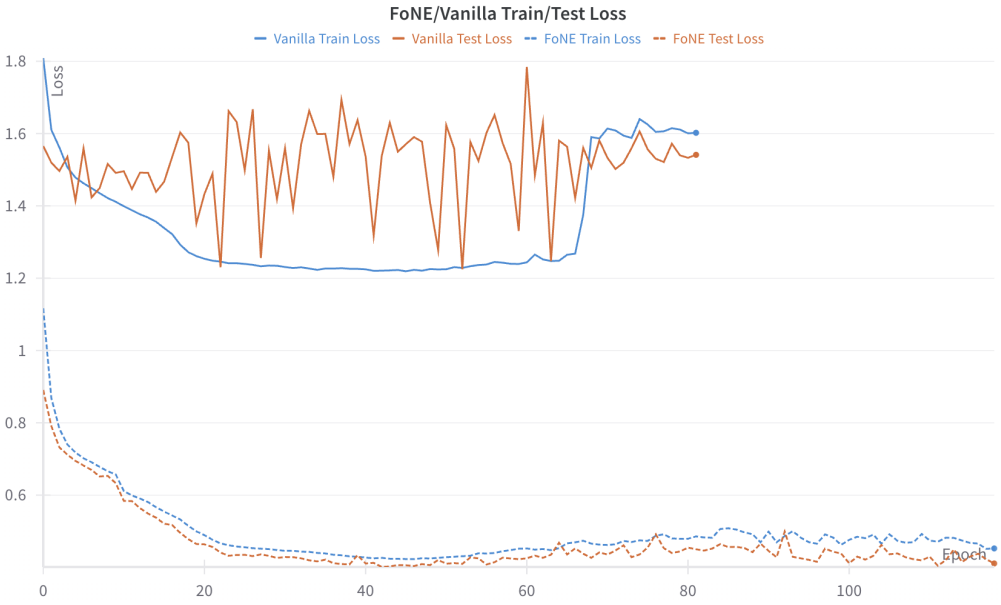


(b) Training and validation losses on the multiplication task for FoNE.

Figure 4: Performance on the multiplication task with FoNE.



(a) Number and digit accuracies on the division task for both FoNE and Vanilla methods.



(b) Training and validation losses on the division task for both FoNE and Vanilla methods.

Figure 5: Performance on the division task between base GPT-2 small and FoNE extension.

6 Conclusion

This study demonstrates that Fourier-based number embeddings (FoNE) improve numerical reasoning by providing structured numerical representations even in smaller transformer models such as GPT-2 small. Fourier analysis of pre-trained GPT-2 embeddings revealed that numerical structures naturally emerge during large-scale pretraining, with distinct frequency peaks indicating an implicit encoding of numerical relationships. However, our own analysis of MLP and attention layers did not necessarily affirm that MLPs capture magnitude estimation, though attention layers roughly saw refined computations through modular arithmetic.

By incorporating FoNE [6], we observed faster convergence and improved arithmetic performance, particularly in addition tasks, demonstrating that structured numerical embeddings facilitate more efficient learning compared to conventional tokenization. However, more complex operations such as multiplication and division remain challenging, suggesting that additional enhancements, such as hierarchical number representations or multi-step reasoning, may be necessary for further improvements.

Beyond arithmetic, the results suggest that Fourier-like structures are intrinsic to pre-trained language models, highlighting broader applications of structured embeddings in numerical tasks such as symbolic computation, scientific modeling, and financial forecasting. Future work remains to be done in investigating more broad symbolic computations that are present inherently in LLMs, and operations beyond the basic four.

7 Contributions

7.1 Anthony Tong

Anthony worked with the FoNE repository code and adjusted it to fit the various experiments ran, writing code to generate the addition and division dataset from scratch, and managed the FoNE addition, multiplication, and division experiments. He also set up the initial testing on base GPT-2 with a training pipeline for the experiment in Section 3.1. Finally, he wrote a significant portion of the introduction, related work, FoNE methods, results, and discussion sections.

7.2 Gavin Simmons

Gavin worked with the GPT2 baseline model, implementing the from scratch model, and the division baseline experiments, creating a different dataset generation script, and changing how the model generated the outputs to allow for multiple token results. He also added a function to be able to track inference time costs for each of the embeddings. Finally, he wrote the discussion for the first paper, the Fourier analysis, and the abstract as well.

7.3 Daniel Warren

Daniel worked to add a configuration file system to the training repo, integrate WandB logging to track different runs, including generating example tables and running validation and test sets. He also trained larger models that would take too long to train on datahub. As well, Daniel implemented the fourier analysis code that visualizes the gradual change in logits over the length of the model and the projections into fourier space.

7.4 Chi Zhang

Chi worked on code for experimenting with the model on different arithmetic operations such as subtraction. Additionally, she wrote a portion of the methods, results, and conclusion sections in the final report, detailing the experimental setup, findings, and implications of the study.

7.5 Christopher Rebollar-Ramirez

Chris helped develop the subtraction dataset generator script and was also involved in training the baseline model on subtraction. Additionally worked on and tested a function to track inference time costs for each embedding. He also contributed to the methods and discussion sections of the paper.

7.6 Acknowledgements

We thank TA Keyu Long and Professor Garrison W. Cottrell for their feedback on our project proposal. Additionally, we thank Professor Robin Jia from USC for delivering an inspiring talk in a seminar in Fall 2024 which inspired this project. Furthermore, we used the publicly-available FoNE code at the repository https://github.com/KevinZhoutianyi/FoNE/tree/master for our FoNE experiments, modifying small portions as needed.

8 References

References

[1] Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit cot to implicit cot: Learning to internalize cot step by step, 2024.

[2] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. Tora: A tool-integrated reasoning agent for mathematical problem solving, 2024.

[3] Nayoung Lee, Kartik Sreenivasan, Jason D. Lee, Kangwook Lee, and Dimitris Papailiopoulos. Teaching arithmetic to small transformers, 2023.

[4] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

[5] Tianyi Zhou, Deqing Fu, Vatsal Sharan, and Robin Jia. Pre-trained large language models use fourier features to compute addition, 2024.

[6] Tianyi Zhou, Deqing Fu, Mahdi Soltanolkotabi, Robin Jia, and Vatsal Sharan. Fone: Precise single-token number embeddings via fourier features, 2025.