Number Embeddings of Pre-trained LMs are Remarkably Accurate

Anonymous ACL submission

Abstract

While language models show excellent capacity to model coherent text, it is commonly believed that their limitations reside in tasks requiring exact representations, such as numeric values. This work shows that representations of numbers that encode their *numeric values* naturally emerge in text-only language models. Contrary to previous work assuming linearity of models' representations, we find that different pre-trained models consistently learn highly precise sinusoidal representations within the input embedding, and can be *accurately decoded* with an appropriate probing method. These findings undermine existing assumptions about the inherent inability of language models to represent numeric information accurately and, consequently, point to the real limitation of robust arithmetic proficiency in language models in their limited capacity to *combine* accurate input representations.

1 Introduction

The landmark paper of Brown et al. (2020) showed that generic neural networks trained on text prediction alone could develop surprising arithmetic capabilities. In the years since, this observation has flourished into a large and vibrant field interested in the arithmetic reasoning capabilities of Transformers (Ahn et al., 2024), rife with research opportunities ranging from interpretability work (Akter et al., 2024) to solving Olympiad-level problems in mathematics (Li et al., 2025). Yet this work has also underscored the limitations of LLMs on arithmetic tasks: Previous studies have explored how models can benefit from incorporating precise numeric representations (Feng et al., 2024), or offloading the arithmetic computation to a tool (Schick et al., 2023; Kadlčík et al., 2023), suggesting that their native learned representations are not reliable. Other works (Kantamneni and Tegmark, 2025; Zhou et al., 2024) have inspected such learned representations directly and tried to understand how models use them. Although model probing methods showed some success in interpreting numeric values from model representations (Zhu et al., 2025), the accuracy of those methods is low, suggesting that learned representations are highly imprecise.

In this paper, we push back on this interpretation: we show that a probe with the *right kind of inductive bias* can retrieve numeric information from number embeddings with *near-perfect accuracy* across an extensive range of LMs, spanning the Llama 3 (Grattafiori et al., 2024) and OLMo 2 (OLMo et al., 2025) series and ranging from 1B to 72B parameters. Given that number embeddings usually follow a sinusoidal wave-like pattern (Nanda et al., 2023; Kantamneni and Tegmark, 2025), this characteristic must be accounted for when designing probes.

We further show how these insights can be leveraged to improve performances on arithmetic reasoning: errors on addition and subtraction tasks can often be matched with an inability of the probe to retrieve the expected numerical information for a given embedding, and demonstrate that intervening on number embeddings such that they more cohesively follow the pattern of other number embeddings can directly improve arithmetic performances. Lastly, we document edge cases that do not fall within this previously understood pattern: in particular, OLMo2 32B (OLMo et al., 2025) learns embeddings that are not sinusoidal-like, despite a high success rate on arithmetic tasks.

2 Related Work

One line of work focuses on incorporating numerical values directly into token representations, providing LMs with a prior. Charton (2022) explores different number encodings based on scientific notation for training LM solvers of linear algebra problems. Golkar et al. (2023) propose represent-

ing numbers as a learned <NUM> token scaled by the number scalar value, demonstrating how models can adopt this scheme for regression tasks.

Another line of work investigates how models learn to represent and process numerical information. Nanda et al. (2023) show that a transformer with one-hot encoding trained from scratch on modular addition discovers Fourier basis and its computation is interpretable in trigonometric functions. Kantamneni and Tegmark (2025) discover an analogous circuitry for (non-modular) addition in a general pretrained language model, and find that its intermediate representations combine both linear and periodic components, reminiscent of a helix structure. Zhou et al. (2024) further identifies subcomponents of the addition circuitry implemented by the attention mechanism and feedforward layers. Zhu et al. (2025) demonstrate that hidden states of pretrained language models can be approximately decoded with a linear (or multi-layer) probe to estimate the logarithm of the number value. Although the probe outputs correlate with the target value, decoding achieves low accuracy.

In summary, prior works suggest that language models *attempt* to encode numerical information into token representations during pretraining, but their precision is rather limited. However, we hypothesize that this perception stems from inadequate probing methods, and learned representations are much more precise than previously estimated.

3 Recovering numerical information from number embeddings

We study LMs from the Llama 3 (Grattafiori et al., 2024), Phi 4 (Abdin et al., 2024), and OLMo 2 (OLMo et al., 2025) series, ranging from 1B to 72B parameters. Wide selection allows us to verify the validity of our observations across a panel of models sharing the characteristic of representing all integers between 0 and 999 with unique tokens.

Motivations. The central and foremost point to address is whether the embeddings representing specific numbers in LLMs contain the numeric information of the value they represent. In practice, this is best addressed with a *probing* setup: If embeddings do contain numerical information, we should be able to learn a decoding function from number embedding to the corresponding integer value. Probing as a methodology comes with its own set of caveats: probes should be kept as simple as possible, and their expressivity should be compared against baseline benchmarks (Hewitt and Liang, 2019). Our specific use case adds further constraints: in particular, we have only one instance per LLM of each integer representation, viz., there is only one vector for the token `42`. This rules out naive classifier implementations, as we aim for the probe to generalize to entirely unseen classes.

Probe architectures. We consider four probes:

$$f_{\text{lin}}(\mathbf{x}) = \mathbf{a}^T\mathbf{x} + b \tag{1}$$

$$f_{\log\ \text{lin}}(\mathbf{x}) = \exp\left(\mathbf{a}^T\mathbf{x} + b\right) - 1 \tag{2}$$

$$f_{\text{sin}}(\mathbf{x}) = (\mathbf{W}_{\text{out}}\mathbf{S})^T(\mathbf{W}_{\text{in}}\mathbf{x}) \tag{3}$$

$$f_{\text{bin}}(\mathbf{x}) = (\mathbf{W}_{\text{out}}\mathbf{B})^T(\mathbf{W}_{\text{in}}\mathbf{x}) \tag{4}$$

where $\mathbf{a}$, b, $\mathbf{W}_{\text{in}}$, and $\mathbf{W}_{\text{out}}$ are learned parameters, whereas $\mathbf{S}$ and $\mathbf{B}$ are means of injecting inductive biases in the linear classifiers f_{sin} and f_{bin}:

$$\mathbf{S}_{ij} = \begin{cases} \sin(ie^j 1000/d) & \text{if } j \equiv 0 \mod 2 \\ \cos(ie^{j+1} 1000/d) & \text{if } j \equiv 1 \mod 2 \end{cases}$$

$$\mathbf{B} = \begin{bmatrix} 0 & \dots & 0 & 0 & 1 \\ 0 & \dots & 0 & 1 & 0 \\ 0 & \dots & 0 & 1 & 1 \\ \vdots & & & & \end{bmatrix}$$

I.e., the i^{th} row of $\mathbf{B}$ corresponds to the integer i expressed in binary, whereas $\mathbf{S}$ is defined as a Fourier basis, suggested by Zhou et al. as the hidden structure learned by pretrained models. The matrices $\mathbf{S}$ and $\mathbf{B}$ thus allows us to partition the label projection of the classifier into three components: a learned projection $\mathbf{W}_{\text{in}} : \mathbb{R}^d \to \mathbb{R}^h$ to project the number embeddings into a reduced low-dimensional space, a fixed matrix ($\mathbf{S}$ or $\mathbf{B}$) allowing us to encode integers using an *a priori* scheme, and a learned projection $\mathbf{W}_{\text{out}} : \mathbb{R}^d \to \mathbb{R}^h$ mapping these *a priori* representations onto the same space as the reduced embeddings. Intuitively, W_{in} uncovers the underlying hidden structure of the learned embeddings, while W_{out} expresses it in terms of interpretable a priori basis, which allows us to generalize to unseen tokens.

Implementation. We evaluate the probes in Equations (1) to (4) using a cross-validation setup with 20 folds. We report their accuracy measured by rounding the output of the regression probes Equations (1) and (2) to the nearest integer, or by retrieving the index of the row in $\mathbf{S}$ or $\mathbf{B}$ that maximizes the output distribution of the classifier

probes Equations (3) and (4). We control the validity of our probes by ensuring that they reach an accuracy of 0 for standard Gaussian vectors as well as for a random permutation of the embeddings. Parameters for regressions are estimated using a least-squares algorithm; whereas our classifiers' parameters are optimized with Adam with a learning rate of 0.0001, weight decay of 0.001, and $\beta = (0.9, 0.999)$. We choose a hidden dimension of 100. The classifiers are optimized to distinguish output *only between training tokens*, and during testing, must choose between all tokens. The probes are optimized until loss converges on a validation split separate from the testing split.

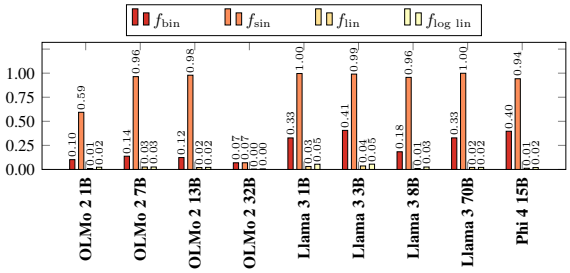


Figure 1: Overview of probes' accuracy (↑).

Results. We summarize performances, measured in terms of accuracy, in Figure 1. Crucially, we are almost systematically able to retrieve the integer value corresponding to the embedding's number with very high accuracy. Another salient observation is that f_{sin} consistently outperforms all other probe architectures including the regression probe used in previous work of Zhu et al. (2025), contradicting their finding that LMs learn to encode numbers linearly. Explaining the success of the Fourier basis, we note that other prior literature has suggested that sinusoidal features are used for arithmetic computation in LMs (Zhou et al., 2024). Adding onto this, we can also stress that, qualitatively, most of the models' whose number embeddings we survey here exhibit wave-like patterns in a PCA projection and have sparse Fourier transform, confirming regularity in the hidden structure. See Figures 3 and 4 in Appendix A.1 for visualizations of PCA and its Fourier transform. Notably, OLMo 2 32B is the only model with low resemblance of the pattern, which is consistent with the low performance of its sinusoidal probe.

4 Leveraging numerical information from number embeddings

Motivations. Having established that number embeddings do encode retrieval numerical information about the integers they represent, we now turn to **how this numerical information is leveraged** by LLMs to perform arithmetic tasks. We study the zero-shot performances of a subset of our models on addition and subtraction tasks. We define our addition task as taking any pair of integers x_1, x_2 such that $0 < x_i < 500$ as input, and computing the expected output $x_1 + x_2$. The subtraction task is defined by taking as inputs any pair x_1, x_2 such that $0 < x_2 < x_1 < 1000$, and computing the expected output $x_1 - x_2$.

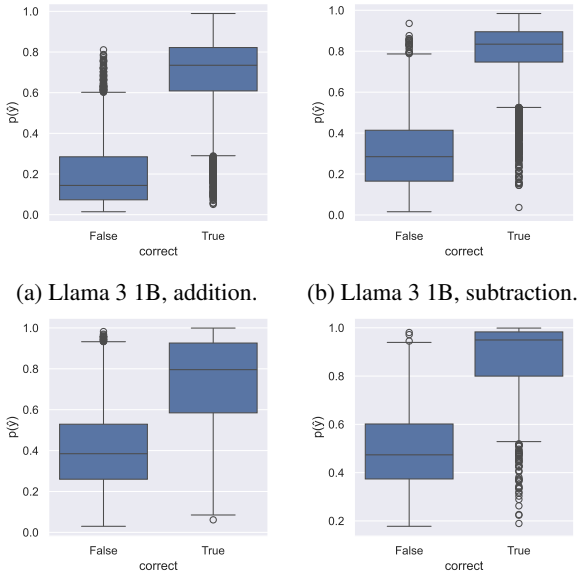
Performances. To perform the arithmetic tasks, we conduct minimal prompt engineering: we systematically evaluate a handful of natural language prompts for their accuracy in a zero-shot setting on the 1B models, and then select the highest-performing for subsequent analyses. All prompts are listed in Appendix B, see Table 4a for addition and Table 4b for subtraction.

	OLMo2 1B	OLMo2 7B	OLMo2 13B	OLMo2 32B	Llama 3 1B	Llama 3 3B	Llama 3 8B	Phi4 15B
Add.	22.21	1.12	0.21	0.05	2.58	0.45	0.24	0.00
Sub.	28.08	0.36	0.17	0.37	1.42	0.15	0.01	0.00

Table 1: Overview of error rates (%, ↓) on arithmetic tasks in zero-shot setting.

An overview of the error rates from the LLMs we study is listed in Table 1. As is apparent, most models achieve high degrees of performance (except for OLMO 2 1B); we also observe a trend towards fewer errors for models with more parameters.

Model behavior. To better explain the behavior of the LLMs, we conduct a simple circuit analysis and a feature attribution experiment using integrated gradients (Sundararajan et al., 2017). For convenience, we focus on the two smaller models in our panel. OLMo 2 1B and Llama 3 1B. Both experiments suggest one major difference between operand pairs leading to failure and to success: the probability assigned by the LLM to the predicted output token tends to be statistically lower when the model produces an incorrect output, see Figure 2. We also observe the same subset of heads being activated for failure and success on the arithmetic task; further details are available in Appendix A.2. Besides the usefulness of this difference in probability mass for diagnostic purposes, these experiments also suggest a difference in *degree* rather than *kind* between failures and successes.



(a) Llama 3 1B, addition. (b) Llama 3 1B, subtraction.

(c) OLMo 2 1B, addition. (d) OLMo 2 1B, subtraction.

Figure 2: Probability mass on the predicted output token when the LLM yields a correct vs. incorrect answer.

	OLMo2 1B	OLMo2 7B	OLMo2 13B	Llama 3 1B	Llama 3 3B	Llama 3 8B
Add.	39.12	21.34	41.22	4.61	1.24	8.01
Sub.	9.06	29.51	26.83	0.39	1.04	0.00

Table 2: Proportion of errors on arithmetic tasks involving misrepresented numbers (%).

Error analysis. To assess how numerical information and arithmetic reasoning are linked, we evaluate whether the errors we see in these arithmetic tasks can be linked to defects of the number embeddings used as inputs. This entails verifying for every pair of operands x_1, x_2 that leads to failure whether either of x_1 or x_2 also leads to a failure in the most accurate probes set up in Section 3, viz. the $f_{\sin}$ probes. The proportion of errors that involve operands not well captured by the $f_{\sin}$ probe varies greatly, depending on the model and task; ranging from more than 40% of all erroneous pairs to 0%; percentages tend to be higher for models that the $f_{\sin}$ probe did not capture well. This suggests that the quality of representations is a factor impacting arithmetic capabilities in models.

Direct intervention. Finally, we test whether intervening on embeddings that our probe fails to capture can improve zero-shot performances on arithmetic tasks. In practice, we start from the $f_{\sin}$ probe described in Equation (3) and trained for OLMo 2 1B fitted on one cross-validation fold, freeze all parameters, and then perform gradient descent to optimize the *embeddings'* of incorrectly decoded tokens with respect to the probes decoding loss. We then replace all embeddings in OLMo 2 1B in the addition input range, namely tokens 55, 117, and 295 with updated embeddings. We finally measure how the replacement impacts error rates. Table 3 summarizes the number of errors on the addition task for pairs of operands involving the corresponding three integers before and after intervention: error rates systematically decrease. This experiment, while of an anecdotal scale, suggests that we can directly intervene on defective number embeddings to align them with the learned hidden structure and observe an improvement in arithmetic performance.

Token	Before	After
55	42	31
117	95	67
295	260	179

Table 3: Error rates before / after a direct intervention on three tokens with high associated addition error rates.

5 Conclusion

In this paper, we have inspected the embedding representations for number tokens across a range of widely used open-source LLMs.

Our observations consolidate a growing body of studies showcasing how LLMs learn sinusoidal hidden structure in number representations. However, building upon this observation, we design a probing method leveraging this structure that decodes LMs' embeddings with *near-perfect* accuracy across multiple models, thus disputing and largely pushing lower-bound estimates of the quality of numeric representations in LMs shown in previous work. Still, we find a model (OLMo 2 32B) that deviates from this pattern, calling into question the generalizability of the conclusions of works such as Zhou et al.'s (2024). Finally, we explore how the knowledge of the structure of numeric information can be exploited by transfers: we show that the preciseness of embeddings relative to reference sinusoidal embeddings can explain a proportion of practical errors on arithmetic tasks. Moreover, we show it is possible to improve accuracy on arithmetic tasks by retrofitting imprecise embeddings by pushing them closer to the model's optimal sinusoidal pattern.

Limitations

Our work, while demonstrating the remarkable accuracy of number embeddings in pre-trained language models, comes with several limitations that warrant consideration for future research.

First, our probing method, though highly effective for many models, relies on an assumed hidden structure of models' learned representations, and therefore expects a priori knowledge of models' mechanics. This necessarily limits the applicability of our approach to models where a known structure exists, e.g., it is not applicable for OLMo 2 32B.

Second, our intervention method was performed on a small-scale experiment, and its generalization across a large suite of models remains an object for future work.

Third, even when we do not perform any pre-training of models, reproducing our experiments requires access to computational resources. We estimate that replicating all our results requires around several hundred GPU hours.

While we recognize the ethical risks associated with AI research, given that our paper focuses on fundamentals of internal representations of numbers within pre-trained language models and their immediate impact on basic arithmetic tasks, broader societal ethical concerns like bias, discrimination, privacy, or job displacement are not directly relevant. Our research operates at a fundamental level of understanding how models encode numerical information, rather than exploring their application or misuse in real-world systems with downstream societal consequences.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, and 8 others. 2024. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

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.

Mst. Shapna Akter, Hossain Shahriar, and Alfredo Cuzzocrea. 2024. Towards analysis and interpretation of large language models for arithmetic reasoning. In *2024 11th IEEE Swiss Conference on Data Science (SDS)*, pages 267–270.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Francois Charton. 2022. Linear algebra with transformers. *Transactions on Machine Learning Research*.

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. *Preprint*, arXiv:2410.13857.

Siavash Golkar, Mariel Pettee, Michael Eickenberg, Alberto Bietti, Miles Cranmer, Geraud Krawezik, Francois Lanusse, Michael McCabe, Ruben Ohana, Liam Parker, Bruno Régaldo-Saint Blancard, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. 2023. xval: A continuous number encoding for large language models. In *NeurIPS 2023 AI for Science Workshop*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

John Hewitt and Percy Liang. 2019. Designing and interpreting probes with control tasks. 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 2733–2743, Hong Kong, China. Association for Computational Linguistics.

Marek Kadlčík, Michal Štefánik, Ondřej Sotolář, and Vlastimil Martinek. 2023. Calc-x and calcformers: Empowering arithmetical chain-of-thought through interaction with symbolic systems. In *Proceedings of the The 2023 Conference on Empirical Methods in Natural Language Processing: Main track*, Singapore, Singapore. Association for Computational Linguistics.

Subhash Kantamneni and Max Tegmark. 2025. Language models use trigonometry to do addition. *Preprint*, arXiv:2502.00873.

Zenan Li, Zhaoyu Li, Wen Tang, Xian Zhang, Yuan Yao, Xujie Si, Fan Yang, Kaiyu Yang, and Xiaoxing Ma. 2025. Proving olympiad inequalities by synergizing llms and symbolic reasoning. *Preprint*, arXiv:2502.13834.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2025. 2 olmo 2 furious. *Preprint*, arXiv:2501.00656.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessí, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: language models can teach themselves to use tools. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

Tianyi Zhou, Deqing Fu, Vatsal Sharan, and Robin Jia. 2024. Pre-trained large language models use fourier features to compute addition. In *Advances in Neural Information Processing Systems*, volume 37, pages 25120–25151. Curran Associates, Inc.

Fangwei Zhu, Damai Dai, and Zhifang Sui. 2025. Language models encode the value of numbers linearly. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 693–709, Abu Dhabi, UAE. Association for Computational Linguistics.

A Supplementary visualization

A.1 Wave-like patterns in embeddings

Figure 3 displays the sinusoidal patterns in Llama 3 70B and OLMo2 13B after re-alignment and detrending with PCA. For clarity, we only include the first 16 principal components.

A.2 Explainability plots for arithmetic tasks.

In Figure 5, we present an overview of head-level attribution of the logits in Llama 2 1B. The same heads in Layers 13 through 15 appear activated in all cases, playing the same inhibitor and booster roles. Incorrectly performed addition leads to a noisier overall pattern. Remarkably, we observe that activity occurs in the latter stages of the model, whereas input embeddings (layer 0) already contain precise numeric information, as per our probing experiments. This delayed processing may explain some of the errors we observe, despite the high accuracy of our probes in Section 3.

B Experimental details

"x_1+x_2 equals to "
"The result of x_1+x_2 is "
"The result of x_1 plus x_2 is "
"The result of x_1 plus x_2 = "
"The result of x_1 plus x_2 ="
"x_1 plus x_2 equals to "
"x_1+x_2="
"x_1 plus x_2 equals "
"x_1 plus x_2 is equal to "
"x_1+x_2 equals "
"x_1+x_2 is equal to "
"x_1 plus x_2 equals "
"x_1 plus x_2 is equal to "

(a) Prompts considered for addition task. x_1 and x_2 are placeholders for the augend and the addend. Prompts are delimited by double quotes; trailing white-space is significant.

"The result of x_1 minus x_2 is "
"The result of x_1 minus x_2 = "
"The result of x_1 minus x_2 ="
"x_1 minus x_2 equals to "
"x_1-x_2="
"x_1 minus x_2 equals "
"x_1 minus x_2 is equal to "
"x_1-x_2 equals "
"x_1-x_2 is equal to "
"x_1 minus x_2 equals "
"x_1 minus x_2 is equal to "

(b) Prompts considered for subtraction task. x_1 and x_2 are placeholders for the minuend and the subtrahend. Prompts are delimited by double quotes; trailing white-space is significant.

Table 4: Prompts considered for engineering of arithmetic zero-shot setting.

C Disclosure of usage of AI assistance

We disclose that we used AI assistance during implementation of this work and its writing. Specifically, we used AI-based code auto-completion (Github Copilot) for increasing productivity of programming, and conversational chatbots (OpenAI ChatGPT, Google Gemini) for improving grammar and fluency of the text. We guarantee that all content is original and factually accurate.

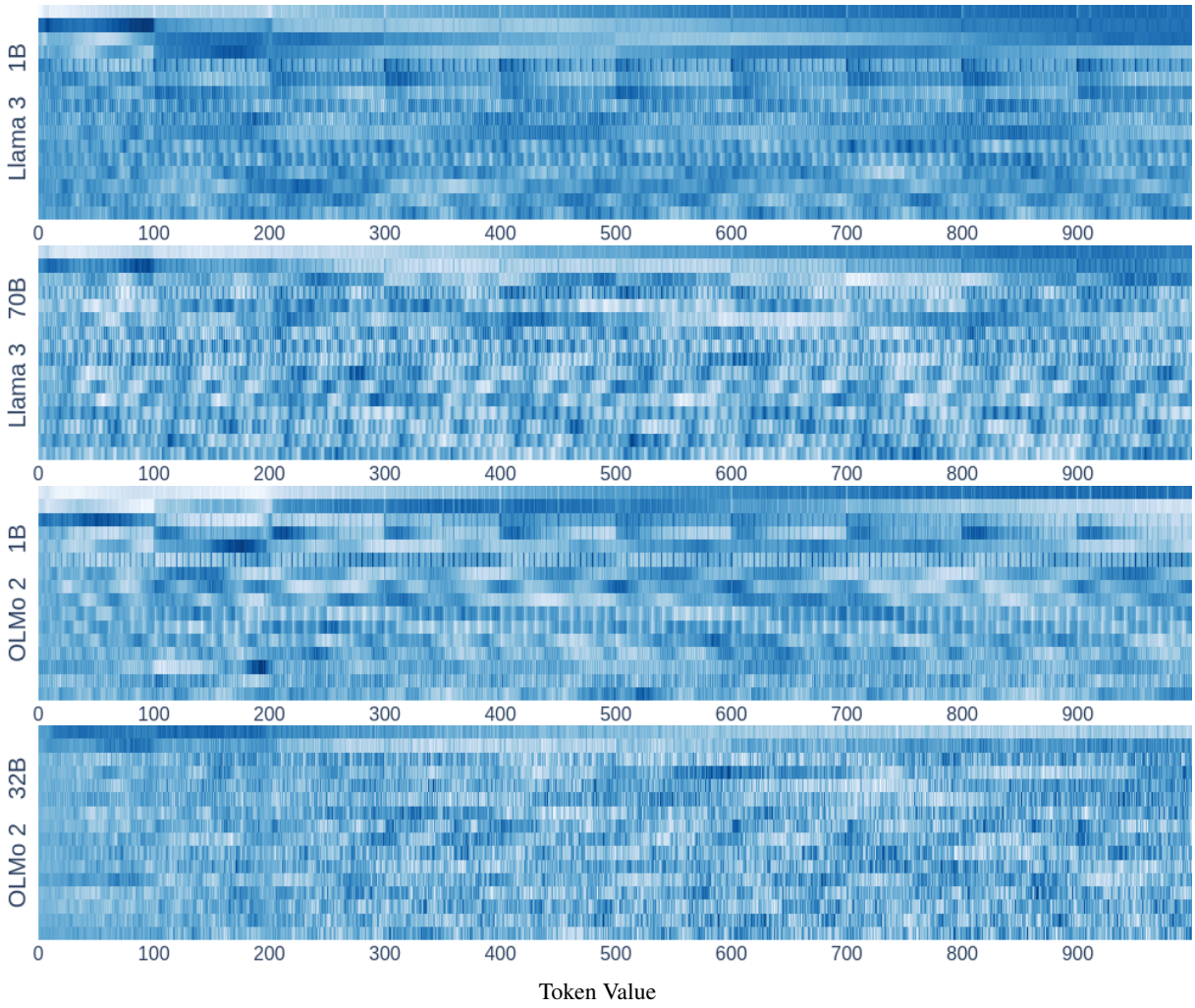
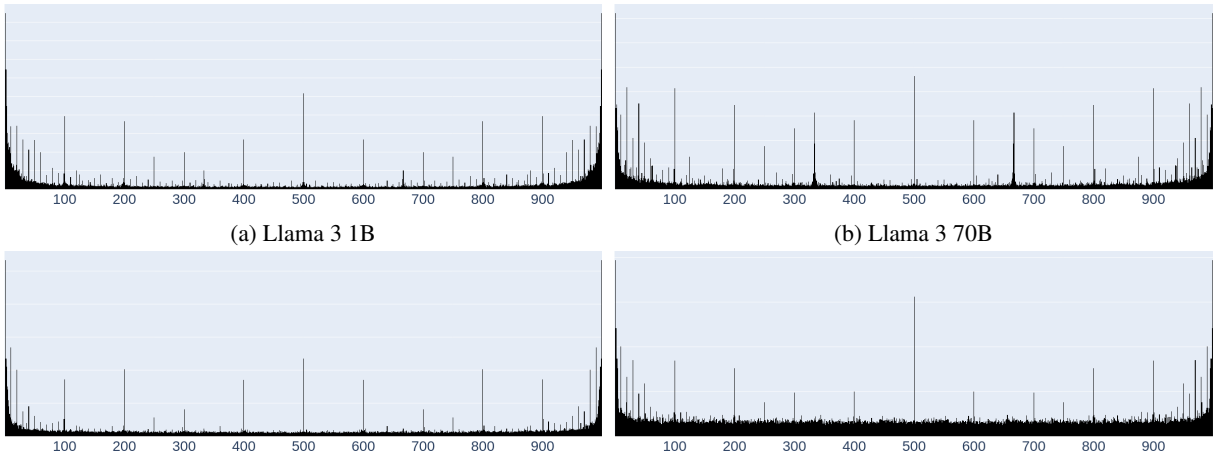


Figure 3: Visualization of PCA (DIM=16) reduced number embeddings, selected models. Although most model exhibit relatively regular wave-like patterns, OLMO 2 32B exhibit little regularity.



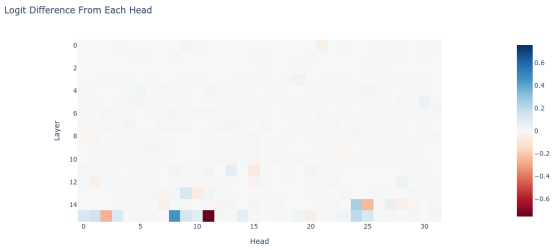
(a) Llama 3 1B

(b) Llama 3 70B

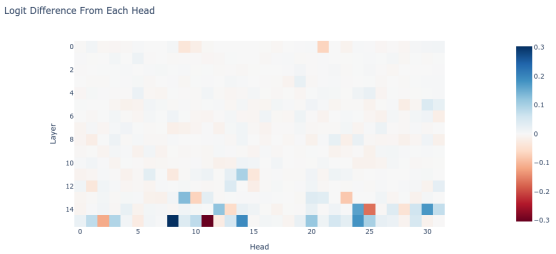
(c) OLMo 2 1B

(d) OLMo 2 32B

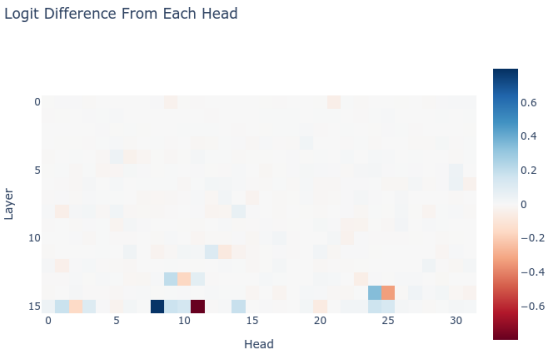
Figure 4: Maximal contribution (magnitude) of each Fourier base frequency's to embedding features in PCA (d=128) reduced space. Sparsity in this plot indicates strong regularity in the hidden structure of model embeddings. OLMo 2 32B has noticeably stronger contribution of all low-contribution frequencies, indicating high irregularity.



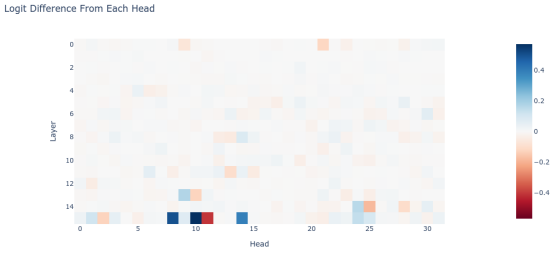
(a) Llama 3 1B, addition performed correctly.



(b) Llama 3 1B, addition performed incorrectly.



(c) Llama 3 1B, subtraction performed correctly.



(d) Llama 3 1B, subtraction performed incorrectly.

Figure 5: Head activations across arithmetic tasks for Llama 3 1B, broken down by task (addition and subtraction) and success (correct or incorrect computation.