Language Models Learn Universal Representations of Numbers and Why You Should Care

Michal Štefánik [*1 2 3] **Timothee Mickus** [*2] **Marek Kadlčík** [*3] **Bertram Højer** [4] **Michal Spiegel** [3 5] **Raúl Vázquez** [2] **Aman Sinha** [6] **Josef Kuchař** [3] **Philipp Mondorf** [7 8] **Pontus Stenetorp** [1 9]

Abstract

Prior work has shown that large language models (LLMs) often converge to accurate input embedding for numbers, based on sinusoidal representations. In this work, we quantify that these representations are in fact strikingly systematic, to the point of being almost perfectly universal: different LLM families develop equivalent sinusoidal structures, and number representations are broadly interchangeable in a large swathe of experimental setups. We show that properly factoring in this characteristic is crucial for assessing how accurately LLMs encode numeric and other ordinal information, and that mechanistically enhancing this sinusoidality can also lead to reductions of LLMs' arithmetic errors.

1. Introduction

Mathematical problems provide an ideal playground in which to assess the current ability of large language models (LLMs) to manipulate precise information. While attempts to foster and improve mathematical reasoning in LLMs have been met with genuine empirical successes, e.g., as an aid to solving mathematical problems (Li et al., 2025), this field is still rife with challenges, particularly in ensuring that LLMs tackling mathematical problems remain robust and reliable. This calls for a deeper understanding of how robust the internal mechanisms of LLMs really are in operating over mathematical information.

[1]R&D Centre for Large Language Models, National Institute of Informatics, Japan [2]University of Helsinki [3]TransformersClub @ Faculty of Informatics, Masaryk University [4]IT University of Copenhagen [5]Kempelen Institute of Information Technology [6]IECL-ATILF, Université de Lorraine - ICANS Strasbourg [7]MaiNLP, Center for Information and Language Processing, LMU Munich [8]Munich Center for Machine Learning (MCML) [9]University College London. Correspondence to: Michal Štefánik <https://michal-stefanik.github.io>.

Mechanistic Interpretability Workshop at the 43rd International Conference on Machine Learning, Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

Towards this goal, prior work identified that number representations stand out in terms of their sinusoidal structure (Nanda et al., 2023; Zhou et al., 2024; Levy & Geva, 2025; Kadlčík et al., 2025), accuracy of input embeddings (Kadlčík et al., 2025), and systematicity of internal mechanisms in addition (Kantamneni & Tegmark, 2025; Zhou et al., 2024). While these works have provided pivotal insights as to how LLMs represent and manipulate numbers, their scope is limited: Kadlčík et al.'s findings only pertain to input embeddings; Zhou et al. find similarity of representations across models but do not quantify their alignment; Levy & Geva focus only on decimals as one particular aspect of representations. Crucially, *no* prior work considers *natural language* contexts, which strongly curtails the generalizability and practical usefulness of their findings, needed for the development of tools able to interpret – and verify – LLMs in practice.

To address these gaps and create a holistic picture of how language models handle numbers, this paper presents two main findings. First, we quantify that sinusoidal structures are **almost perfectly universal** across LLMs: models of different sizes and families converge to sinusoidal representations that are universal across *models* (§3.1) and *layers* (§3.3) while remaining highly systematic and *accurate* throughout the model's internal processing (§3.2) – showing it is possible to track the processing of numeric information throughout models.

Second, we demonstrate the necessity of **suitable explainability tools** to make sense of how LLMs process numbers: we show how accounting for the sinusoidal nature of representations allows us to unravel the mechanism of manipulating large, multi-token numbers (§4.1) as well as other ordinal data (§A.9) in LLMs, and we also provide a mechanistic evidence that a precision of sinusoidality in the model's internal representation determines its output accuracy across arithmetic tasks (§4.2).

In summary, our main contributions are:

1. We develop a new, more accurate probe of numeric values that generalizes well across different tasks and models

2. We quantify that numeric representations of numbers are sinusoidal and highly accurate *throughout the model*, beyond the input embedding representations;
3. We are the first to analyze the LLMs' numeric processing in *natural-language* settings and show that training numeric probes with real-world data is crucial for creating generalizing, practically useful probes;
4. We investigate mechanisms of numeric manipulations beyond addition and provide a mechanistic evidence that sinusoidality is crucial for models' accuracy.

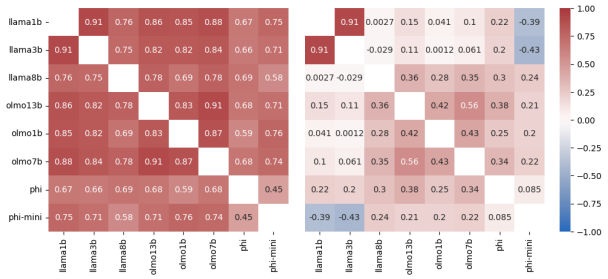
We make the new probe, with all our methods and analyses publicly available for any use.[1]

2. Related work

Mathematical capabilities of LLMs The seminal findings of Brown et al. (2020) on remarkable arithmetical abilities of LLMs have ushered in a large body of work aimed at evaluating more complex mathematical capabilities of LLMs (Hendrycks et al., 2021; Cobbe et al., 2021; Sun et al., 2024; Yu et al., 2024) or lack thereof (Nikankin et al., 2025). This in turn has encouraged a focus on how to bolster the capabilities of LLMs or interpret how LLMs perform computations (Zhang et al., 2024; Stolfo et al., 2023). Most relevant to ours, some of the previous work focuses specifically on how numbers should be represented in principle. For instance, Charton (2022) assesses an impact on models performance in linear algebra problems when employing different encoding schemes based on scientific notation on linear algebra problems. Golkar et al. (2023) propose to encode numeric values by incorporating a scaled, learned control token <NUM>. Feng et al. (2024) remark that the precision of the numeric representation type impacts performance on arithmetic tasks, and argue to use the logarithmic-precision architecture of Feng et al. (2023).

Numeric representations in LLMs The increased focus on evaluating LLM performance in verifiable domains such as math problems has also led to increased interest in the mechanisms by which LLMs compute arithmetic functions. Nanda et al. (2023) demonstrate how a model trained from scratch on modular addition relies on trigonometric operation: it maps inputs onto a unit circle, corresponding to a specific rotation, and then learns to combine the two rotations to derive a valid solution. Kantamneni & Tegmark (2025) apply circuit analysis to understand how general pretrained models perform addition, and report that representations are mapped onto a helix that can be manipulated using trigonometric operations. Zhou et al. (2024) refine these observations in qualitative analysis of Fourier components in LLMs' representations (note that Fourier transformations map arbitrary functions into combinations of periodic trigonometric functions), and highlight the distinct role of attention and feedforward sublayers. Focusing more narrowly on representations rather than processing, Zhu et al. (2025) argue that non-linear probes do not provide a better fit than linear probes. Levy & Geva (2025) remark that it is possible to retrieve the digit (value mod 10) of numeric inputs with high accuracy. Our work builds upon Kadlčík et al. (2025) that propose a probe architecture that factors in the sinusoidal nature of representations, and show that they can retrieve numeric values from input embeddings with much higher accuracy.

[1] https://github.com/prompteus/numllama



(a) Cosine-based RSA for number tokens

(b) Cosine-based RSA for random tokens

Figure 1. RSA scores for input embeddings of LLMs

3. Universality of Sinusoidal Representations of Numbers

3.1. Different Models Learn Equivalent Representations of Numbers

The first point we address is the distinctiveness of LLMs' representations of numbers. Results from Kadlčík et al. (2025) and Zhu et al. (2025) suggest that different models converge to the same *type* of number representations, but they do not provide a direct assessment of how *closely* the representations match across models – which could evidence that a shared representation is a causal consequence of architectural bias and optimization process rather than a coincidental artifact. Following Kadlčík et al., we focus on the input embeddings from eight LLMs of diverse sizes and families with open-sourced, pre-trained checkpoints: OLMo 2 (OLMo et al., 2025), Llama 3 (Grattafiori et al., 2024) and Phi 4 (Abdin et al., 2024).

To quantify whether models converge to similar embeddings, we start by conducting a simple Representational Similarity Analysis (RSA; Kriegeskorte et al., 2008) across the input embeddings for number word-pieces in paired models. As a baseline, we compute RSA scores for a random sample of 1000 pieces present in the vocabularies of all the models under consideration. Results of the analysis are displayed in Figure 1. Across all pairs of models, we find that number

embeddings systematically yield consistently higher RSA scores than a random sample of word pieces. While some models (especially in the Phi family) are not as well aligned with other models, we do observe high scores both within and across families, demonstrating that models converge to embedding spaces with **equivalent similarity structure**.

Another approach to quantifying the similarity of number representations is to inspect their base frequencies through Fourier decompositions (Kadlčík et al., 2025; Zhou et al., 2024). Applying a PCA transformation followed by a Fourier transform allows us to quantify the *magnitude* of individual frequencies (Figure 2a). Subsequently, we determine whether two models agree as to how they rank frequencies by computing a simple intersection-over-union (IoU) of the top k frequencies. We repeat this process for every pair of models, using the input embeddings of number pieces (Figure 2b) as well as that of random pieces (Figure 2c). When considering the top $k = 63$ frequencies, we find **perfect agreement across all models** for numbers. This provides a quantitative basis for some of Zhou et al.'s (2024; §C.1) qualitative observations.[2]

In summary, our analyses show that the LLMs from different families learn to represent numbers in the same topology and trigonometric features using the same dominant frequencies. This is specific to numbers: we do not reproduce these observations with other word pieces shared across models.

3.2. Number Representations are Always Sinusoidal

In Section 3.1, we presented evidence that the input embeddings of numbers *across* different LLMs are highly consistent. We now focus on whether the sinusoidal mechanism of representations is maintained and employed *throughout* the model.

For qualitative assessment, we visualize the PCA and its Fourier transform of internal activations of numerical tokens in Section A.4. Besides the apparent wave-like pattern, the representations have a sparse Fourier transform, confirming the sinusoidal character. For quantitative analysis, we look at the model's internal representations through the lens of the sinusoidal probe proposed by Kadlčík et al. (2025). This probe was designed to map input embeddings of LLMs into an integer, thus classifying embeddings into predefined range of numeric values. It is defined as:

$$f_{\sin}(\mathbf{x}) = (\mathbf{W}_{\text{out}}\mathbf{S})^T(\mathbf{W}_{\text{in}}\mathbf{x})$$
$$\mathbf{S}_{ij} = \begin{cases} \sin(ie^j 1000/h) & \text{if } j \equiv 0 \pmod 2 \\ \cos(ie^{j+1} 1000/h) & \text{if } j \equiv 1 \pmod 2 \end{cases} \quad (1)$$

where $\mathbf{W}_{\text{in}} : h \times d$ and $\mathbf{W}_{\text{out}} : h \times d$ are learned parameters, and $\mathbf{S} : h \times 1000$ injects an inductive bias in the classifier towards sinusoidality.[3]

While Kadlčík et al. demonstrate that such a probe properly benefits from the expected sinusoidal inductive bias and greatly bolster accuracy for number representations, their approach does not enforce a strict sinusoidal structure and may fail to generalize due to redundancy in a model's representation. As an alternative, we design a new probe which assigns a *single sine wave to each latent feature* and *directly* optimize the frequency, phase-shift, and amplitude of each sine wave:

$$f_{\text{param-sin}}(\mathbf{x}) = \mathbf{S}^T(\mathbf{W}_{\text{in}}\mathbf{x})$$
$$\mathbf{S} = \mathbf{w}_a \sin(\mathbf{w}_\phi + \mathbf{w}_f \mathbf{r}^T) \quad (2)$$

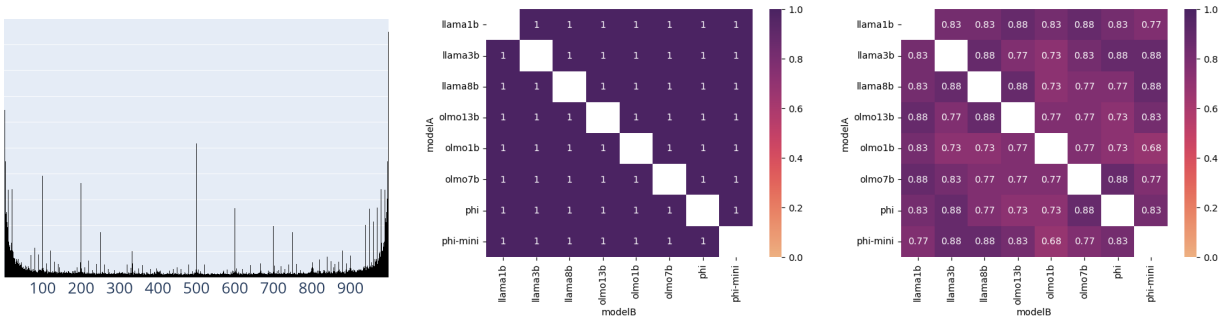
where $\mathbf{r}$ are constant label values (0...999 for all models we analyze), and $\mathbf{w}_f, \mathbf{w}_a, \mathbf{w}_\phi \in \mathbb{R}^d$ are three vectors of learnable parameters modeling the frequencies, amplitudes and the phase-shifts of the sine waves. This architecture *strictly constrains* the classification matrix $\mathbf{S}$ to take the form of d sine waves sampled at n evenly-spaced points. The configuration of initialization and hyperparameters is provided in Appendix B.

We first assess whether sinusoidal probes are the most suitable, i.e., an accurate choice for decoding the *internal* representations of the model. To contextualize the accuracy in terms of sinusoidal quality of the representation, we also evaluate other types of probes used to decode numbers in previous work as baselines; using logarithmic (Feng et al., 2023) or linear base (Zhu et al., 2025), or sinusoidal features as targets (Kadlčík et al., 2025). We fit separate probes for activations from every layer across six different language models from three families. Contrary to previous work, we assess the representations of numbers in *natural-language* contexts, rather than as single tokens or synthetic arithmetical commands. Our contexts cover four domains (temporal, arithmetic, medical and culinary) covering numerical contexts including quantities, measurements or timestamps (detailed in Appendix A.2). To assess generalization rather than memorization capacity, we *split* the prompts and corresponding representations into training, validation and test sets containing *distinct* extracted numbers (x_2).

Figure 3a shows the accuracy of different probes extracting the value of input numbers from representations of each model's layer, evaluated across six different models. As we can see, both sinusoidal probes reach over 80% of accuracy in all but two cases and over 90% of accuracy for a majority of probing scenarios. Here, the superior accuracy of both sinusoidal probes provides an indicator of both (i) the sinusoidal character, and (ii) the accuracy of input number representation across layers — the best probe results present

[2] See our Section A.1 for technical details, including how the value of k is set.

[3] Unless otherwise stated, we use $d = 100$; h corresponds to the inner dimensionality of the LLM at hand.

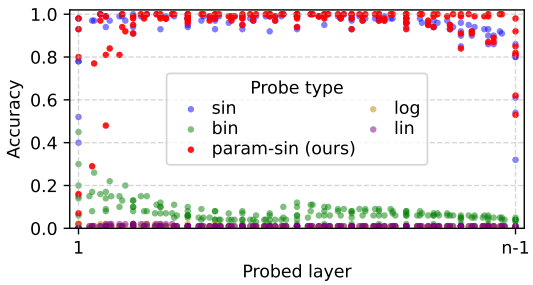


(a) Contributions of Fourier base frequencies (shown example: Llama-3 1B)

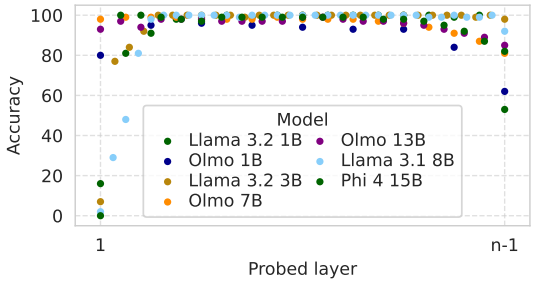
(b) IoU for number pieces

(c) IoU for random pieces

Figure 2. Intersection-over-union of top $k = 63$ Fourier base frequencies



(a) using probes from prior work with different bases.



(b) across different language models; param-sin probe.

Figure 3. **Accuracy of decoding numbers** from internal activations corresponding to numeric tokens

a lower bar of the accuracy of numeric representation on a given layer.

Figure 3b disentangles the accuracies of the param-sin probe from Figure 3a by the model, covering six LLMs of different sizes and three different families. While the probing is slightly less accurate on some particular layers of particular models, for none of the analyzed models does the sinusoidal probe systematically underperform. If we take into account that our probes need not be perfect and reflect a *lower bound* of real accuracy of numeric representations, these results taken together show that a sinusoidal representation is a decisively dominant scheme of numeric representations in language models, corroborating and extending the observations of Kantamneni & Tegmark (2025).

A vast majority of cases performing lower than 80% occurs in probing the models' first and a last layer; in Section A.6, we analyze these cases in detail through the lens of probes learned representations and find the substantial qualitative difference in the *sparsity* of representations of numbers. While the middle layers of all models resort to a 'dense' representation decomposed into a small set of features bearing full numeric information, the initial and final layers scatter the numeric information across most of the latent features. As the scattered features share the information with natural language, a robust recovery in a realistic setting of natural contexts becomes more difficult.

Generalization across contexts We can see that sinusoidal probes trained with natural-language contexts can provide highly accurate estimates of LLMs' numerical states across models. However, favoring simplicity, all of the previous work (§2) investigated numerical representations in *synthetic* mathematical contexts. To assess whether such a simplification comes for a price in practical applicability of resulting probes, we explore whether the choice of training contexts affects generalization of resulting probes.

In Figure 4, we compare the performance of probes trained on natural-language contexts (yellow) and synthetically-constructed math contexts (blue), evaluated on natural-language inputs (solid lines) and synthetic inputs (dashed lines). We can see that probes fitted with natural-language contexts are much more robust in application across *both* math and natural-language context. While both probes generalize well within their training context type, **only the natural-language probes generalize across evaluation settings**, robust to the context's distribution shift. This observation has two important implications. First, it informs future work aiming to explain models' internal computations in practical applications to favor probes trained in natural-language settings. Second, it restrains future work

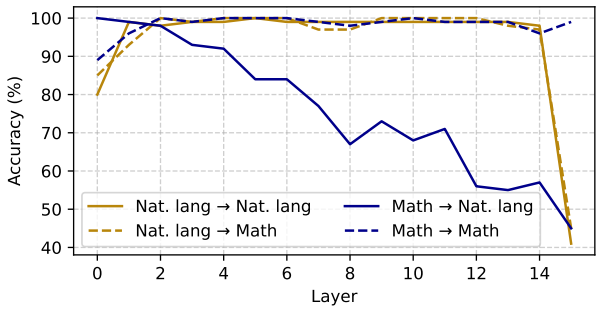


Figure 4. **Generalization of probes** fitted on representations in natural language (yellow), and synthetic, mathematical contexts (blue), evaluated on natural-language (solid) and math contexts (dashed). Llama 3.2 1B.

in interpretability and explainability from drawing general conclusions on models' mechanics from fully-synthetic settings, evidencing that such conclusions may not generalize to real-world applications.

In summary, our results underscore that the same type of sinusoidal representation of numbers holds in general across different model types. Numbers are represented in a similar, systematic fashion, regardless of which layer or type of context we consider. While we can expect this to be in part due to the residual stream across layers, we note that there is no a priori reason for an LLM to preserve sinusoidality across layers, and previous work shows that input tokens tend to have a negligible impact on upper layer representations (Mickus et al., 2022).

3.3. Different Layers use Interchangeable Numeric Representations

Having found the remarkable degree to which the embeddings for numbers align across different models and the noteworthy universality of sinusoidal representations in varied contexts, as a final step, we turn to assess whether the models utilize the identical representation throughout its computation. To investigate that, we employ the probes from Section 3.2 trained separately for each separate model layer (L_i) and measure the accuracy of each probe on representations derived for every *other* layer $L_j \neq L_i$.

While informative, this approach still comes with the caveat that probes trained for a given layer L_i may pick up on idiosyncrasies inherent to a specific layer — i.e., the probes might not disentangle what is specific to numbers as opposed to what is specific to L_i. To address this point, we also fit probes using *all but one* layer ($L_1, L_2, \ldots L_{i-1},\ L_{i+1}, \ldots L_n$), and evaluate performance on representations from the held out layer L_i. As before, we hold out 100 numbers for assessing generalization in validation and test conditions.

Results of probes' cross-layer generalization for Llama 1B are displayed in Figure 5. We observe that probes fitted on a concrete layer's representations (Figure 5a) generalize outstandingly well to both close-by and more distant layers, with the exception of the first and the last layer. This trend reveals that language models **operate in a consistent representation** of numbers that is **universal throughout their computation**, undergoing only minor shifts across layers. This trend is also corroborated by multi-layer probes evaluated in heldout-layer fashion (Figure 5b), reaching a close-to 100% accuracy across *all* intermediate layers and six models of diverse sizes and families. This result has practical implications for future work in interpretability, showing that we can train universal, yet highly *accurate* probes for hidden layers of a broad set of language models.

To summarize, we find that the same key, i.e. probe can be used to interpret numeric information at different layers of the same language model. Added to our previous observations in Sections 3.1 and 3.2, we can therefore stress that models learn input embeddings for numbers that are **strikingly similar**, regardless of which model they come from. These embeddings are processed into sinusoidal representations that are systematic and consistent regardless of context or hidden layer, with the source of this consistency maintained primarily by the residual stream across layers (cf. Section A.7).

4. How to Study Numbers in LLMs

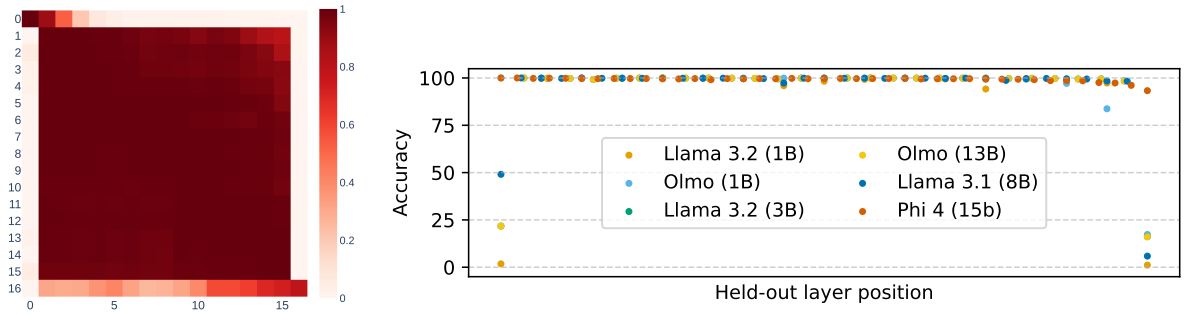
We have established that models universally converge to sinusoidal number representations. This characteristic is not shared with other word pieces. This questions the appropriateness of existing interpretability tools when it comes to numbers and other ordinal data types.

4.1. Multi-token Numbers

As a case study for observing how the choice of interpretability toolset affects our findings, we analyse whether our conclusions about the remarkable accuracy of numeric representations also hold for large, *multi-token* numbers.

First, we use our sinusoidal probes to retrieve the values of *all parts* of multi-token numbers from a *single*, i.e. last token's representation. For this experiment, we adapt the methodology utilizing natural-language contexts (see Figure 4), with a substitution of original numeric values with values spanning two to six numeric tokens, i.e. in a numeric range between $10^3 - 10^{18}$. We use a subset of BBC data that was published after the cutoff or release date of our LLMs (Li et al., 2024).

As a point of comparison, we consider the general-purpose TunedLens tool (Belrose et al., 2025), an early-exit interpretability method which aims to explain the contents of intermediate representations by learning to translate them



(a) accuracy of probes trained for Llama 1B on each single layer (rows) evaluated on other layers (columns)

(b) probes trained on all-but-one layers, evaluated on the held-out layer

Figure 5. **Transferability of representations**: Accuracy of param-sin probe evaluated on **unseen layers**

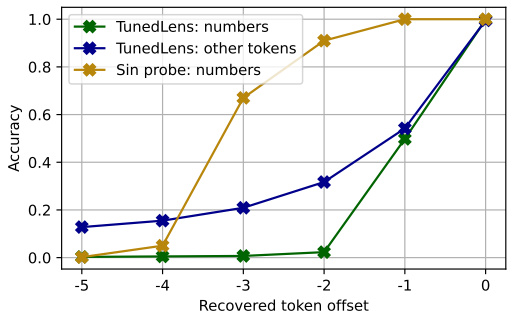


Figure 6. Accuracy of extracting pieces of **multi-token numbers** from the last token representations.

into logits. For each layer L_i, TunedLens distills the computations done in the subsequent layers $L_{i+1}, \dots L_n$ into a simple affine transformation $\mathbf{W}_i$, such that applying this transformation $\mathbf{W}_i$ to a hidden state at L_i followed by the unembedding matrix closely matches the logits that the LLM would eventually produce. TunedLens was applied to number representations in previous work (Zhou et al., 2024) and also represents the line of work arguing for a linear representations of numeric values in LLMs (Zhu et al., 2025). To maximize comparability and align the number and distribution of target categories, we narrow down the space of probed tokens to 1,000 tokens ranked as 2,000–3,000th most-common tokens in our dataset (matching the number of classes and their distribution for our number probes). Crucially, we also report previous-token recovery rates using TunedLens for *number* tokens, using the same configuration as for natural-language tokens.

Results for Llama 1B and different probing methods are presented in Figure 6. Results of sin probes show that **multi-token numbers are indeed systematically and highly accurately *superposed* in the latest numeric representation** – reaching an accuracy of 99% for the immediately-preceding number piece (offset -1). However, the accuracy and/or systematicity of the superposition mechanism quickly drops for numbers longer than *three* tokens (i.e. $\geq 10^9$), reaching close to zero for the 5th preceding token.

We notice that **TunedLens gives a conflicting account**. While it shows that previous non-number tokens are also superposed, i.e. can be decoded as far as five time-steps later with non-zero probability, it suggests that this is not the case for number tokens – these cannot be retrieved past two time-steps, and even then, only with 50% accuracy. However, a comparison with the sin probe shows that this trend is merely a result of TunedLens's inefficiency for numeric tokens; estimates derived from it provide an inaccurate representation of LLMs' accuracy, underestimating the amount of contained information and overlooking the superposition mechanism uncovered by a more suitable probe.

4.2. Output Tracking

A reliable mechanism of numeric manipulations necessitates an accurate representation of inputs (evidenced in §3), but also the ability to construct systematic and accurate representations of *outputs*. Without the specialized tools to analyze numeric representations, prior work describes LLMs' arithmetical mechanisms through qualitative assessments limited to the outputs of addition (Kantamneni & Tegmark, 2025; Zhou et al., 2024; Levy & Geva, 2025). Therefore, having found the sinusoidal probes to be highly suitable for describing the representations of input numbers, we set off to investigate whether such probes can also better capture the mechanics of LLMs' *output* representations, compared to other probe types.

In this set of experiments, we prompt the language models with prompts of the form "*The result of* $x_1 \otimes x_2$ *is* " and train probes to predict the correct result of $x_1 \otimes x_2$, where $\otimes \in \{+, -, \times, /\}$.

Figure 7 shows the results for probes from prior work, including linear probes from Zhu et al. (2025); Zhou et al. (2024) and a sin probe proposed and applied by Kadlčík

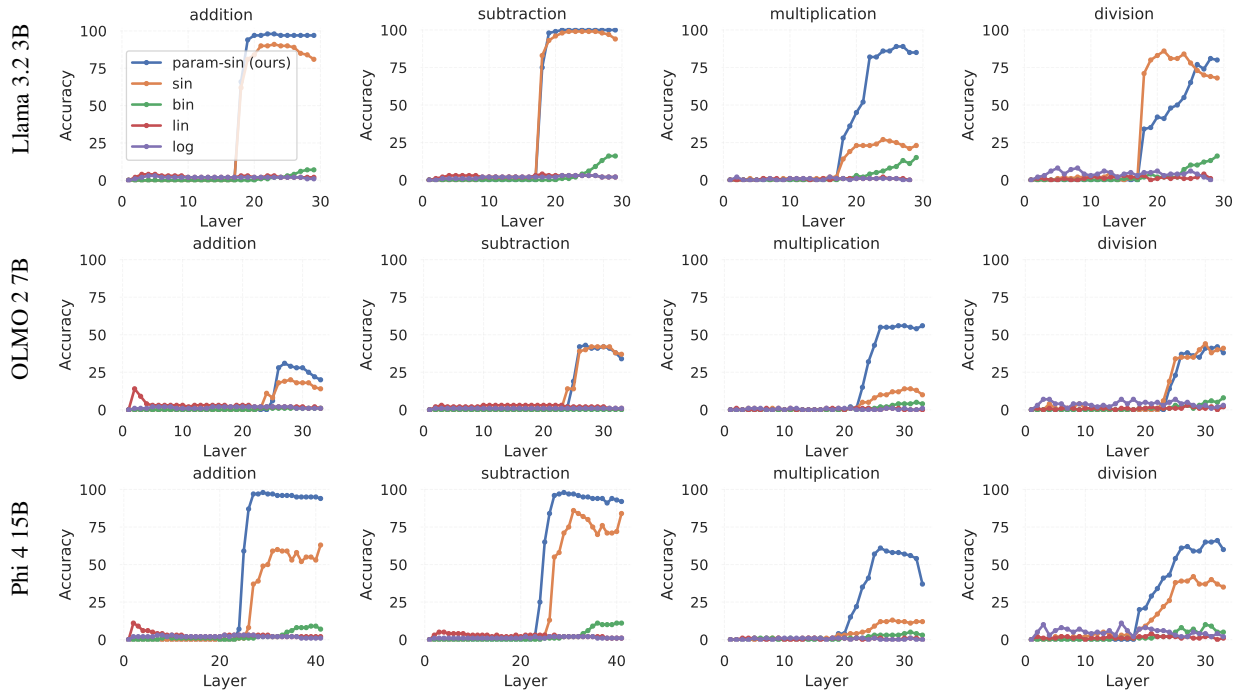


Figure 7. **Probing accuracy** in extracting correct output from internal activations across layers; different probes.

et al. (2025). We can see that the param-sin probe performs comparable or better than probes applied for numbers in previous work and newly allows the tracking of outputs with over 80% of accuracy in multiplication. The results for addition also push the known lower bounds of precision of the result emerging early in the model, conflicting with a hypothesis of "gradually refining predictions" from Zhou et al. (2024) based on LogitLens, but supporting the interpretation of arithmetics implementation in LLMs by Stolfo et al. (2023) through a combination in late layer(s).

Figure 8 further details on a distinction between the cases where the model predicts a correct and incorrect response – suggesting that the **extent to which the model maintains the sinusoidal representation may determine its accuracy**.

We evidence this by following qualitative and mechanistic evidence. In Section A.5, we observe the output embeddings of addition and multiplication – latter coupled with lower model's accuracy – differ in the precision of the the observed sinusoidality; with the pattern being more prominent in the better-performing, addition embeddings. In Section A.8, we further analyze how, according to our probes, is the result refined via gradually minimizing the response error across layers, with particular layers deteriorating this refinement, suggesting a worse fit of the sinusoidal probe and thus, the sinusoidal pattern. We find that simply dropping the layers causing a divergence from the probe's fitted pattern lead to an accuracy improvement in multiplication and division in 4 out of 6 cases.

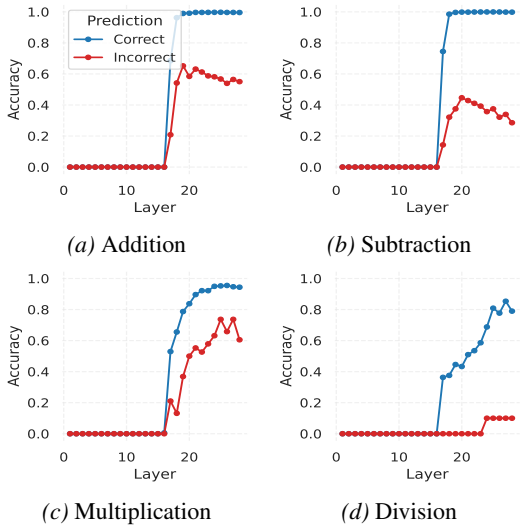
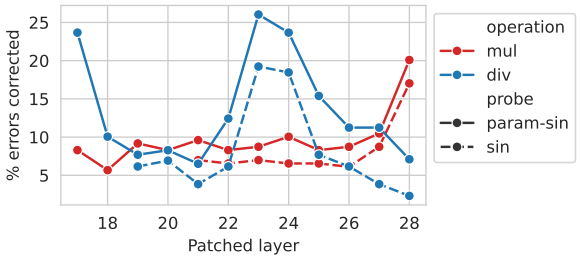


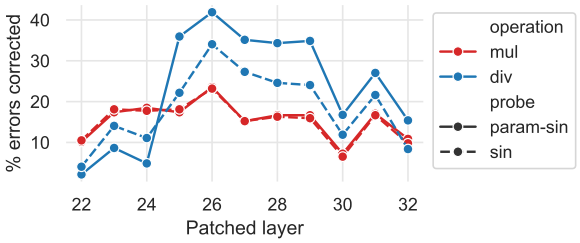
Figure 8. **Probing accuracy** in extracting correct output from Llama 3.2 3B's internal activations separates model's correct (blue) and incorrect (red) predictions.

Finally, we evidence this point more directly by a mechanistic intervention – **steering activations** in prediction cases leading to incorrect prediction **towards the expected sinusoidal pattern**. Concretely, we (i) fit sinusoidal probes at each layer to predict the output of a given arithmetic operation; (ii) optimize a set of 1,000 randomly-initialized 'patch' embeddings $\mathbf{e}_y$ to maximize the probability that the probes will map them onto the integer value y; and (iii) steer

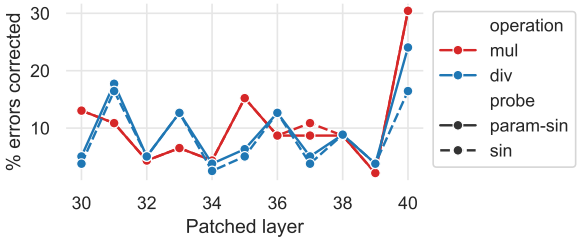
activations towards the corresponding patch whenever the model is not producing the expected output. Steering is achieved by interpolating between the activation $\mathbf{h}_i$ at layer L_i and the patch $\mathbf{e}_y$ for the target scalar y, $\alpha\mathbf{h}_i + (1-\alpha)\mathbf{e}_y$. We report results with respect to the optimal value of α per layer, operation and probe.



(a) Llama 3B



(b) OLMo-2 7B



(c) Phi-4

Figure 9. **Error reduction** after **steering activations** on incorrect predictions of multiplication and division. Optimal interpolation factor α per setup.

Figure 9 displays the absolute difference in performance caused by the steering on previously erroneous cases in multiplication and division for three models. We target probes on upper layers which reach accuracy scores $\geq 10\%$. In short, depending on the layer, we can correct up to 30% of errors for multiplication, and up to 42% for division. Crucially, our probe almost always outperform the probe proposed by Kadlčík et al. (2025): Across all setups, we only observe four setups where the sin probe outperforms our sin-param probe.[4] The optimal setup per model and operation is always obtained with our probe. This underscores once again the importance of selecting the right explanatory tool, this time in the context of activation steering.

In sum, factoring in the knowledge of the sinusoidal nature of number representations allowed us to understand and assess the lower bounds of accuracy of multi-token number representations. On a case study of Llama 3.2 3B, we further showed how more accurate probes allow pinpointing and **eliminating** sources of errors. Tools that are best suited to number representations allow us to enhance accuracy and perform more effective interventions.

4.3. Ordinal Data Types

Beyond pure numbers, we find that our probe generalizes to other ordinal data types, specifically days of the week, months, and spelled-out numerals, even in extreme data scarcity ($n \leq 12$). As detailed in Section A.9, our specialized architecture achieves high decoding accuracy across diverse models, which corroborates findings of Engels et al. (2025) on circular representation of these concepts. Additionally, our probe consistently outperforms the sinusoidal probe of Kadlčík et al. (2025), demonstrating the broader utility of our probing tool.

5. Conclusions

In this paper, we have shown the number representations in LLMs are exceptional in two distinct but related ways. First, the regularity and systematicity of the sinusoidal structure that all LLMs converge is striking, to the point of being almost universal across all experimental setups we consider. We can view the same frequencies in Fourier being used by all models we survey (§3.1), we can train a probe on one layer and apply to another (§3.3). The one remarkable exception is the discrepancy between natural language contexts, which yield remarkably robust probes that generalize broadly, and synthetic probes derived from templated texts, which do not.

Second, this exceptional sinusoidal regularity requires adapting the way we study number processing in LLMs. Using a general-purpose TunedLens method, applied for numbers in prior work, grossly underestimates how accurately previous timestep information can be retrieved, whereas sinusoidal probes such as the one we propose in Equation (2) may lead to substantially higher accuracies and contradictory conclusions (§4.1). Finally, we demonstrate that the sinusoidal probes are also well applicable for analyzing the emergence and manipulation of model's own predictions and that the accuracy of representation in terms of sinusoidality is related to the prediction accuracy (§4.2).

While we focus on numbers, our experiments showcase how important the appropriate interpretability toolset is for accurate interpretation of specialized data types. This naturally questions the common practice of designing *general-*

[4] Viz., division with OLMo at L22, L23, L24), multiplication with Phi at L37.

purpose interpretability and explainability tools, motivating future work towards adapting our practices to the specifics of the type of data we want to explain.

Impact Statement

This paper deepens the understanding of learned representations of pretrained Language Models, advancing the field of Machine Learning. To the best of our knowledge, there are no direct negative ethical or societal consequences of our work.

Acknowledgments

This work is supported by the Research Council of Finland through project No. 353164 "Green NLP – controlling the carbon footprint in sustainable language technology".

This project has received funding from the European Union's (EU) Horizon Europe research and innovation programme under Grant agreement No. 101070350. The work has also received funding from the Digital Europe Programme under grant agreement No 101195233 (OpenEuroLLM). The contents of this publication are the sole responsibility of its authors and do not necessarily reflect the opinion of the EU.

References

Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R., Gunasekar, S., Harrison, M., Hewett, R. J., Javaheripi, M., Kauffmann, P., Lee, J. R., Lee, Y. T., Li, Y., Liu, W., Mendes, C. C. T., Nguyen, A., Price, E., de Rosa, G., Saarikivi, O., Salim, A., Shah, S., Wang, X., Ward, R., Wu, Y., Yu, D., Zhang, C., and Zhang, Y. Phi-4 technical report, 2024. URL https://arxiv.org/abs/2412.08905.

Belrose, N., Ostrovsky, I., McKinney, L., Furman, Z., Smith, L., Halawi, D., Biderman, S., and Steinhardt, J. Eliciting latent predictions from transformers with the tuned lens, 2025. URL https://arxiv.org/abs/2303.08112.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners, 2020. URL https://arxiv.org/abs/2005.14165.

Charton, F. Linear algebra with transformers. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=Hp4g7FAXXG.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Engels, J., Michaud, E. J., Liao, I., Gurnee, W., and Tegmark, M. Not all language model features are one-dimensionally linear. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=d63a4AM4hb.

Feng, G., Zhang, B., Gu, Y., Ye, H., He, D., and Wang, L. Towards revealing the mystery behind chain of thought: A theoretical perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=qHrADgAdYu.

Feng, G., Yang, K., Gu, Y., Ai, X., Luo, S., Sun, J., He, D., Li, Z., and Wang, L. How numerical precision affects mathematical reasoning capabilities of llms, 2024. URL https://arxiv.org/abs/2410.13857.

Golkar, S., Pettee, M., Eickenberg, M., Bietti, A., Cranmer, M., Krawezik, G., Lanusse, F., McCabe, M., Ohana, R., Parker, L., Blancard, B. R.-S., Tesileanu, T., Cho, K., and Ho, S. xval: A continuous number encoding for large language models. In *NeurIPS 2023 AI for Science Workshop*, 2023. URL https://openreview.net/forum?id=KHDMZtoF4i.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Kadlčík, M., Štefánik, M., Mickus, T., Kuchař, J., and Spiegel, M. Pre-trained language models learn remarkably accurate representations of numbers. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Proceedings of the 2025 Conference*

on Empirical Methods in Natural Language Processing, pp. 26693–26702, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1356. URL https://aclanthology.org/2025.emnlp-main.1356/.

Kantamneni, S. and Tegmark, M. Language models use trigonometry to do addition, 2025. URL https://arxiv.org/abs/2502.00873.

Kriegeskorte, N., Mur, M., and Bandettini, P. Representational similarity analysis - connecting the branches of systems neuroscience. *Front. Syst. Neurosci.*, 2:4, November 2008.

Levy, A. A. and Geva, M. Language models encode numbers using digit representations in base 10. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 385–395, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-190-2. doi: 10.18653/v1/2025.naacl-short.33. URL https://aclanthology.org/2025.naacl-short.33/.

Li, Y., Guerin, F., and Lin, C. Latesteval: Addressing data contamination in language model evaluation through dynamic and time-sensitive test construction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):18600–18607, Mar. 2024. doi: 10.1609/aaai.v38i17.29822. URL https://ojs.aaai.org/index.php/AAAI/article/view/29822.

Li, Z., Li, Z., Tang, W., Zhang, X., Yao, Y., Si, X., Yang, F., Yang, K., and Ma, X. Proving olympiad inequalities by synergizing llms and symbolic reasoning, 2025. URL https://arxiv.org/abs/2502.13834.

Mickus, T., Paperno, D., and Constant, M. How to dissect a Muppet: The structure of transformer embedding spaces. *Transactions of the Association for Computational Linguistics*, 10:981–996, 2022. doi: 10.1162/tacl_a_00501. URL https://aclanthology.org/2022.tacl-1.57/.

Nanda, N., Chan, L., Lieberum, T., Smith, J., and Steinhardt, J. Progress measures for grokking via mechanistic interpretability. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=9XFSbDPmdW.

Nikankin, Y., Reusch, A., Mueller, A., and Belinkov, Y. Arithmetic without algorithms: Language models solve math with a bag of heuristics, 2025. URL https://arxiv.org/abs/2410.21272.

OLMo, T., Walsh, P., Soldaini, L., Groeneveld, D., Lo, K., Arora, S., Bhagia, A., Gu, Y., Huang, S., Jordan, M., Lambert, N., Schwenk, D., Tafjord, O., Anderson, T., Atkinson, D., Brahman, F., Clark, C., Dasigi, P., Dziri, N., Guerquin, M., Ivison, H., Koh, P. W., Liu, J., Malik, S., Merrill, W., Miranda, L. J. V., Morrison, J., Murray, T., Nam, C., Pyatkin, V., Rangapur, A., Schmitz, M., Skjonsberg, S., Wadden, D., Wilhelm, C., Wilson, M., Zettlemoyer, L., Farhadi, A., Smith, N. A., and Hajishirzi, H. 2 olmo 2 furious, 2025. URL https://arxiv.org/abs/2501.00656.

Stolfo, A., Belinkov, Y., and Sachan, M. A mechanistic interpretation of arithmetic reasoning in language models using causal mediation analysis. In Bouamor, H., Pino, J., and Bali, K. (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7035–7052, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.435. URL https://aclanthology.org/2023.emnlp-main.435/.

Sun, Y., Yin, Z., Guo, Q., Wu, J., Qiu, X., and Zhao, H. Benchmarking hallucination in large language models based on unanswerable math word problem. In Calzolari, N., Kan, M.-Y., Hoste, V., Lenci, A., Sakti, S., and Xue, N. (eds.), *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 2178–2188, Torino, Italia, May 2024. ELRA and ICCL. URL https://aclanthology.org/2024.lrec-main.196/.

Yu, L., Jiang, W., Shi, H., YU, J., Liu, Z., Zhang, Y., Kwok, J., Li, Z., Weller, A., and Liu, W. Metamath: Bootstrap your own mathematical questions for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=N8N0hgNDRt.

Zhang, W., Wan, C., Zhang, Y., ming Cheung, Y., Tian, X., Shen, X., and Ye, J. Interpreting and improving large language models in arithmetic calculation, 2024. URL https://arxiv.org/abs/2409.01659.

Zhou, T., Fu, D., Sharan, V., and Jia, R. Pre-trained large language models use fourier features to compute addition. In Globerson, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 25120–25151. Curran Associates, Inc., 2024. URL https://openreview.net/forum?id=i4MutM2TZb.

Zhu, F., Dai, D., and Sui, Z. Language models encode the value of numbers linearly. In Rambow,

O., Wanner, L., Apidianaki, M., Al-Khalifa, H., Eugenio, B. D., and Schockaert, S. (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 693–709, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.47/.

A. Supplementary results

A.1. Selecting the optimal number of Fourier components for comparison

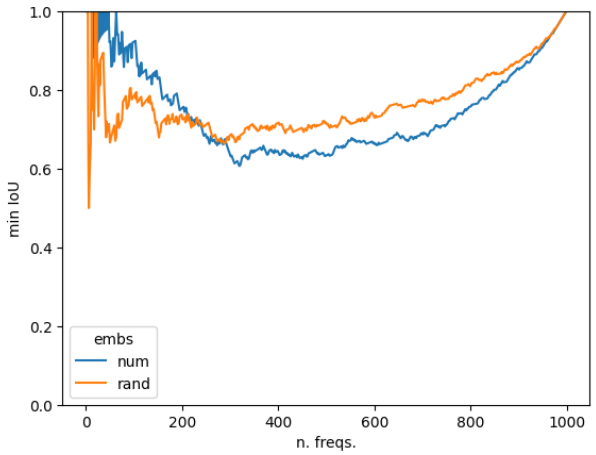


Figure 10. Minimum IoU across number of frequencies considered

In Section 3.1, we discuss selecting the optimal number of Fourier components. As can be assessed from 2a, from a certain frequency rank, Fourier decompositions resort to noise. Thus, our objective is to find the "cutoff" rank (k) that disentangles oscillating information from the non-oscillating, presumably non-numeric information. This means finding an *optimal*, yet non-trivial (e.g. ≥ 10) cutoff for a number of components. In practice, we evaluate for each value of k what the minimum IoU agreement score across all models amounts to for numbers and random overlapping word-pieces. As displayed in Figure 10, we find such an optimal cutoff at $k = 63$ as the highest value of k that leads to perfect agreement across models.

A more thorough assessment of Figure 10 suggests a few interesting trends. Random word pieces also tend to favor a handful of Fourier basis components; which we conjecture is due to the sampling mechanism. By selecting overlapping pieces, our pieces must be frequent enough to be present in multiple distinct tokenizers, which in turns shapes the type of linguistic units represented in this random sample. Secondly, we also observe a 'cross-over' point around $k \approx 250$, after which we find greater agreement in random pieces than numbers. Yet, we note that (i) a significant proportion of the mass is concentrated in a few frequency components for numbers, whereas random pieces lead to much more uniform distributions across frequency components (see Figure 11); and (ii) the results in Figure 10 still allow us to establish that the Fourier profile of number pieces is clearly distinct from what we observe for any other overlapping set of word pieces.

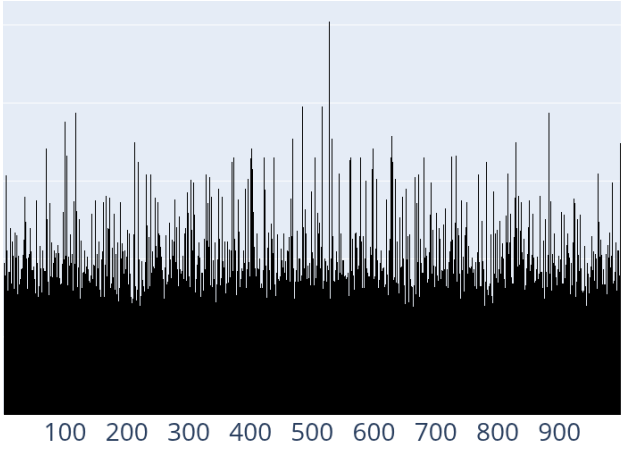


Figure 11. Contributions of Fourier base frequencies for non-number pieces (Llama-3 1B, provided as an example)

A.2. Natural-language contexts for representations extraction

Contrary to prior work, we use the contexts of natural language to extract the representations of numbers. We argue that such setting better corresponds to the practical applicability of probes – in monitoring models' behavior and errors, or in tracking the faithfulness of reasoning chains.

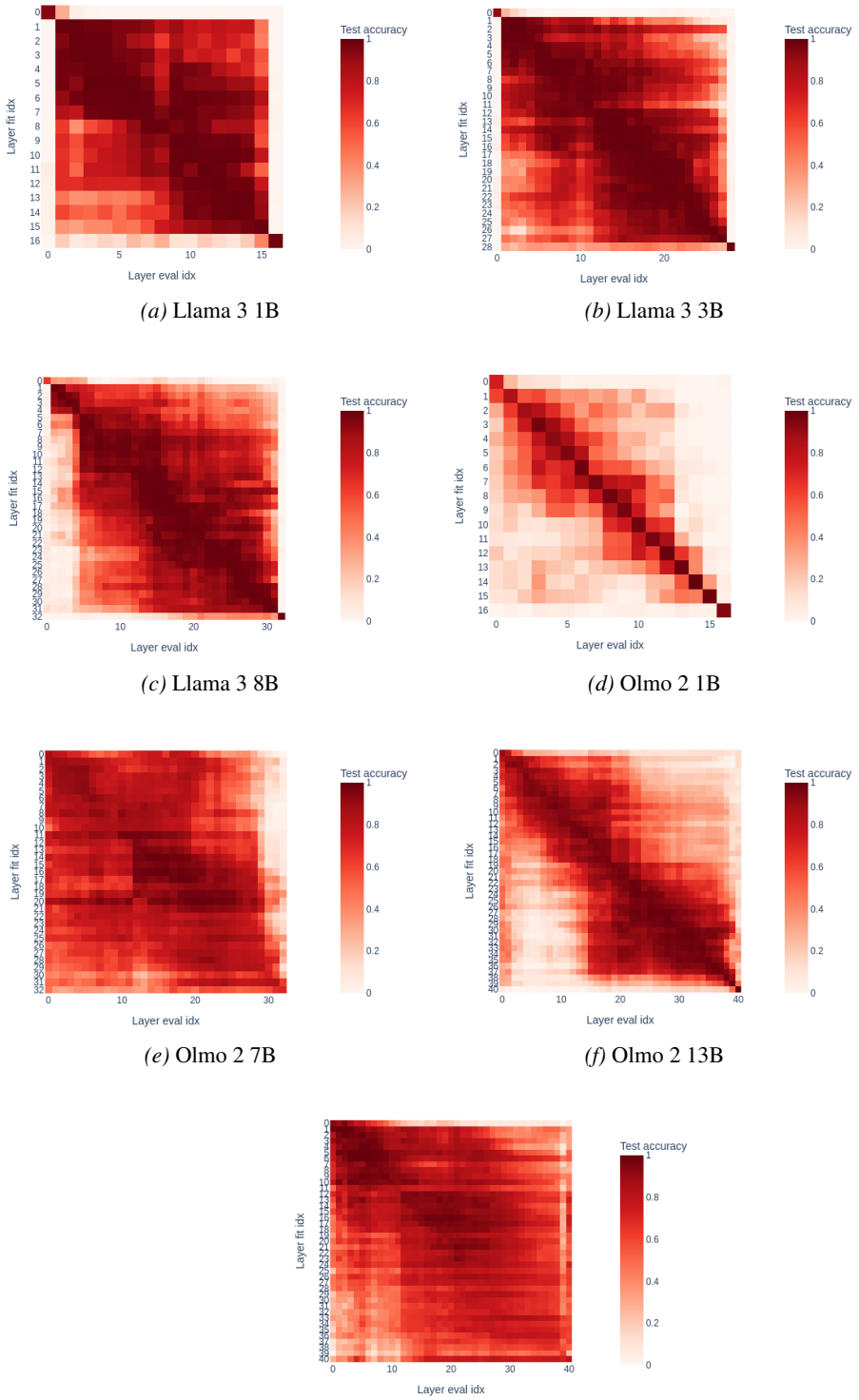
We choose the contexts with medium to high co-dependency on numerical contents covering four distinct domains and eight different datasets. The datasets categorized into these domains are presented in Table 3 and total in around 700 thousands natural-language contexts.

A.3. Generalization of probes to unseen layers

Figure 12 shows how probes fitted on each layer generalize to all other layers in a model, using synthetic mathematical contexts. Strong cross-layer generalization indicates a high consistency of representations throughout the model.

A.4. Visualizations of internal activations

We visualize the internal activations for a string template "x1 + x2 =" on the second numeric token (x2). Values for x2 are selected as a range 0-999 (all values present by the vocab), and values for x1 are sampled randomly from the same range. Then, we project the activations of the layer to 64 dimensions with PCA and compute the Fourier transform.



(a) Llama 3 1B

(b) Llama 3 3B

(c) Llama 3 8B

(d) Olmo 2 1B

(e) Olmo 2 7B

(f) Olmo 2 13B

(g) Phi 4 15B

Figure 12. Probe fitted on one layer evaluated on all layers, synthetic math contexts. Olmo 2 1B shows the weakest cross-layer generalization among all models. Llama models display a strong separation between input/output embedding representations and the hidden representations.

	Add	Sub	Mul	Div
Accuracy	100%	99.8%	90.4%	5.9%
P(Extracted | Incorrect)	-	56.8%	26.3%	94.4%
P(Not extracted | Correct)	1.4%	0.03%	26.1%	5.9%

Table 1. **Probe accuracy in extracting the predicted result:** Ratio of cases where the sin probe of some layer (top) retrieves a correct result when the model's prediction is *incorrect*, and (bottom) can *not* retrieve a correct result when the model's prediction is *correct*.

We visualize the first 16 PCA dimensions in Figure 13 and the maximal magnitudes of the frequencies in the Fourier transform in Figure 14. We visualize across model sizes and families.

A.5. Qualitative assessment of output embeddings

We visualize output representations of Llama 3 1B on addition and multiplication operations. For each expected result value y in the range 0-999, we sample a random pair ($x1$, $x2$) from the same range, such that $x1 + x2 = y$ (or $x1 \times x2 = y$, respectively). We then predict the next token for prompts "$x1$ plus $x2$ is " and "$x1$ multiplied by $x2$ is " and collect the final output representations of the model before decoding. We then reduce the representations with PCA to 16 dimensions and visualize the result in Figure 15.

A.6. Probe sparsity

Cross-layer evaluation indicates that models' internal representations tend to differ from the input/output embeddings. We find a notable discrepancy in the *sparsity* of their representation; Whereas the input/output embeddings represent numbers in a more *distributed* fashion, hidden layers use a small number of consistently-ordered sin features.

This is visualized in Figure 16, showing that intermediate layers in Llama 1B lead to probes with very few weights with values $> 10^{-5}$.

We hypothesize that this discrepancy can be caused by the models' computational and optimization mechanics. As cross-entropy loss is minimized by confident one-hot predictions, models can benefit from using more features to create higher dissimilarity (separability) of tokens. In contrast, internal representations are not subject to direct optimization pressure and might benefit from fewer but more informative features.

Nevertheless, we note that this trend is pertinent across different models to a different extent: OLMo 7B and Phi 15B learn more distributed features also across the intermediate layers — which, in turn, also leads to a better generalization of universal probes (Figure 5b), but still causing an outstanding drop in held-out accuracy compared to other layers.

Finally, we note that the models' sparsity profile does not relate to reported performances on arithmetic tasks (Kadlčík et al., 2025) and it also does not necessarily determine the accuracy of probes trained specifically for particular layer(s) (Figure 3).

A.7. Using probes for tracking representation consistency

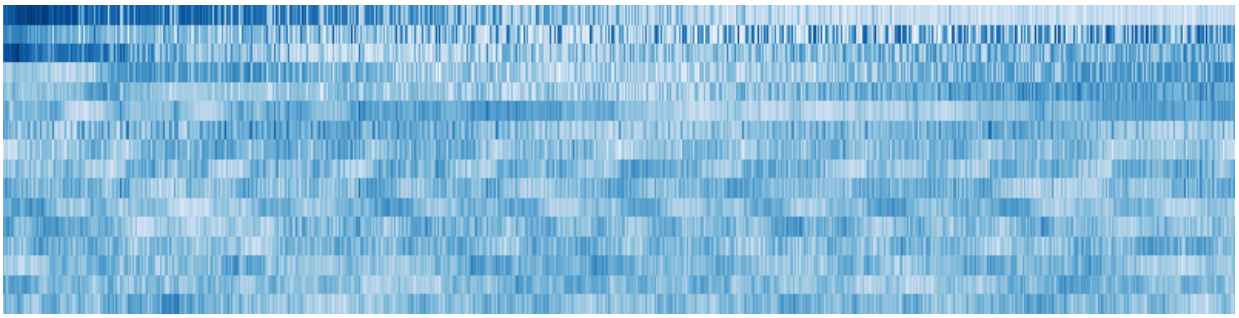
We explore the origin of the high consistency of representations across layers within the internal mechanism that each transformer layer implements. In Figure 17, we visualize the accuracy of probing input representations from each of the components present in transformer layers (Llama 3.2 1B for brevity). We find the accurate numeric representation is scattered across different components in the attention mechanism, with the attention output projection largely violating the sinusoidal representation, which is then, to a large extent, reconstructed in the subsequent fully-connected block. Nevertheless, the consistency of layers' output representation is primarily maintained by residual streams across layers.

A.8. Incremental refinement of model predictions on arithmetic tasks

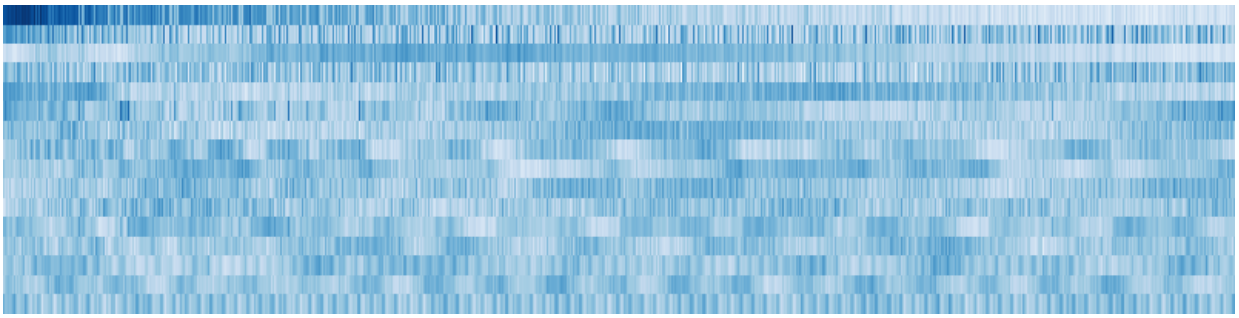
We can use sinusoidal probes to estimate the absolute error per layer: the absolute difference between the predicted and target integer provides an estimate of how far off from the intended target a model is at any intermediate step of computation. Results aggregated per layer of Llama 3.2 3B are displayed in Figure 18. Across all operations, models tend to incrementally reduce the error towards the true answer value, *gradually* refining across layers. These results also suggest that there are particular layers responsible for an increase of errors in the model's internal computation.

Figure 19 corroborates this hypothesis — showing that particular layers 'break' the correct result probed from the previous layer in large proportions of all predictions. For instance, in division, the 5th layer is responsible for breaking the correct result recoverable from the previous layer in over 10% of all cases. Further analyses, presented in Table 1, show that according to the probed representations, the models often achieve a correct result *internally*, even though it does *not* surface: this is the case of 56.8% of all surfaced (prediction) errors in subtraction, 26.3% in multiplication, and as much as 94.4% of all errors in division.

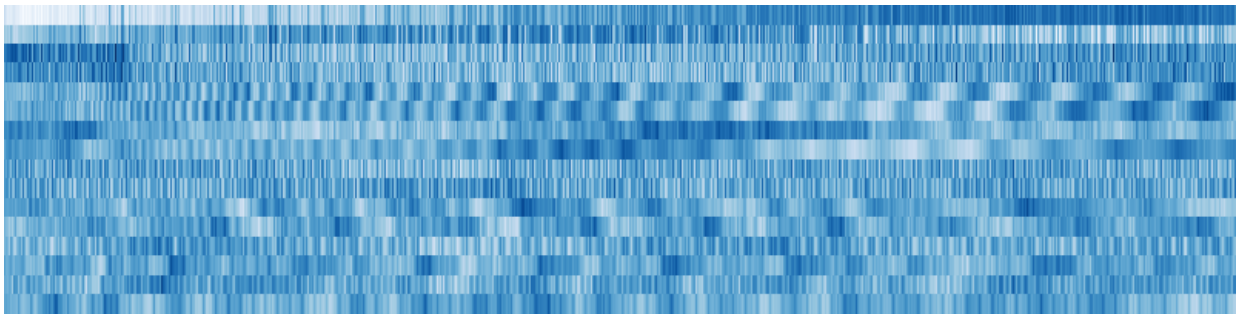
Having used our probes to identify key layers that are likely responsible for errors, we can now turn to practical **interventions** that they make possible. In practice, we simply remove the suspected layers from the model. This extremely naive intervention strategy is justified by the universality of representations across layers (Section 3.3). We remove one of the three layers with the highest error aggregation, separately in multiplication and in division (having a potential



(a) Llama 3 1B (layer 8/17)

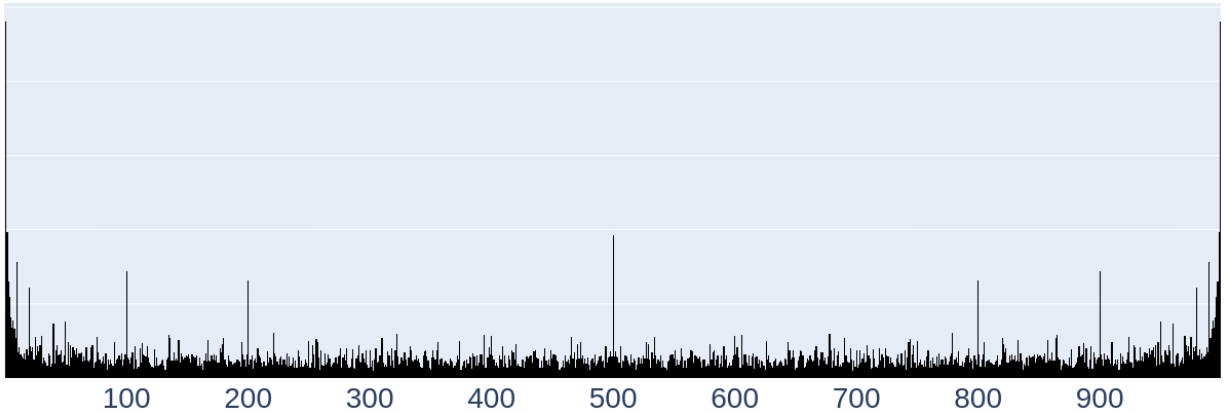


(b) Olmo 2 7B (layer 16/31)

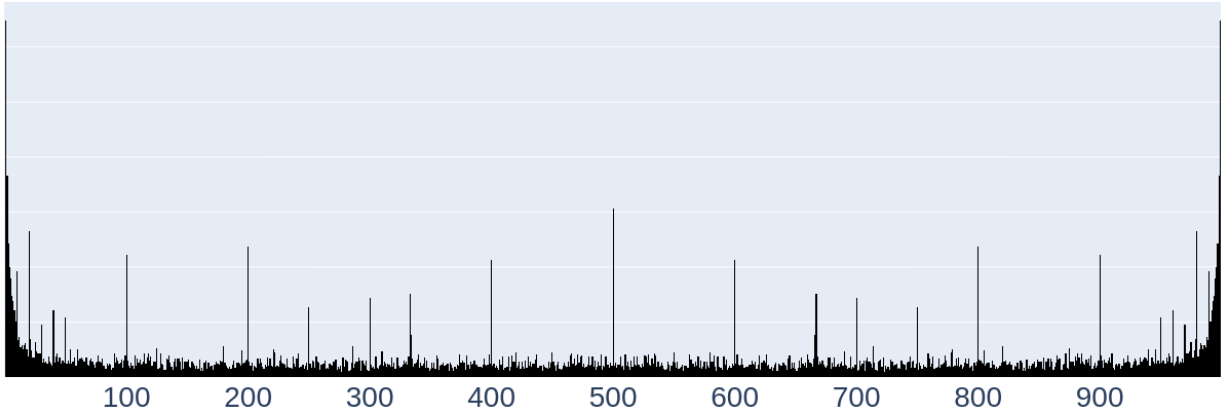


(c) Phi 4 15B (layer 20/41)

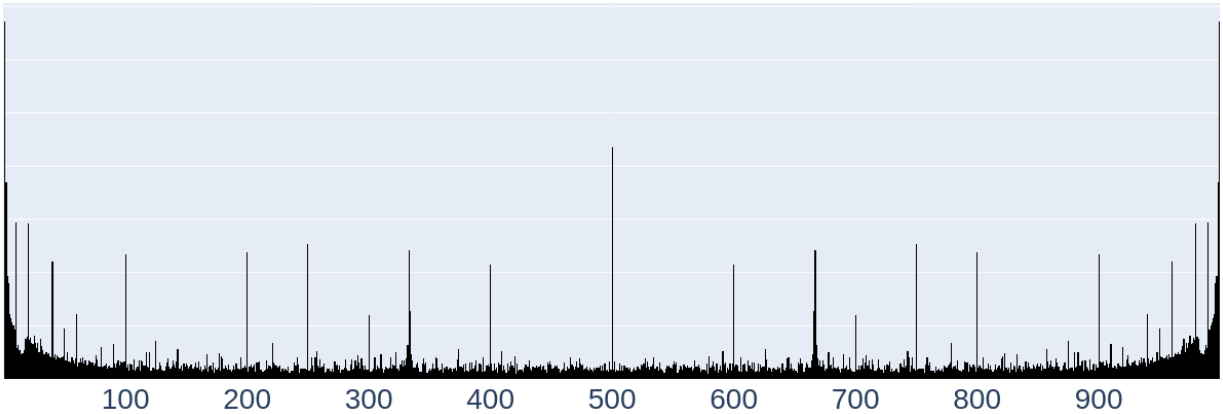
Figure 13. PCA of models' internal representations



(a) Llama 3 1B (layer 8/17)

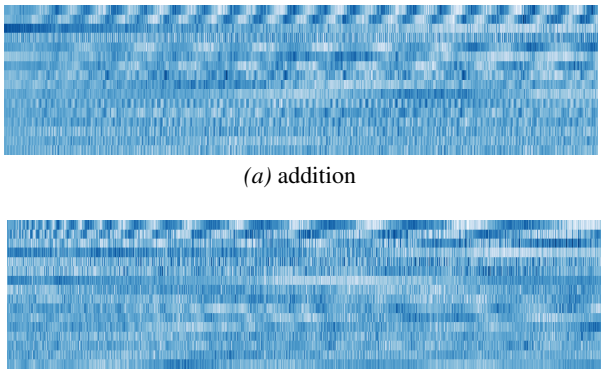


(b) Olmo 2 7B (layer 16/31)



(c) Phi 4 15B (layer 20/41)

Figure 14. Maximal magnitudes of frequencies in Fourier transform of PCA of models' internal representations



(a) addition

(b) multiplication

Figure 15. PCA of Llama 3 1B *output* representations for the operations of addition (recoverable with close-to 100% accuracy) and multiplication (recoverable with appx. 80% of accuracy)

for improvement of accuracy). We find that this enhances performance in *four* out of six cases — in one case (layer 4) in multiplication, causing a 26% error reduction (from 90.38% to 92.91% and in *all* cases in division, with error reduction between 27–64%. However, we note that the absolute number of correct predictions probed in the lower layers is negligible, which hinders the general applicability of this methodology.

A.9. Ordinal Data Types

Having established that sinusoidal mechanisms robustly encode numeric values and our specialized probe facilitates targeted interventions, we investigate how this tooling generalizes to ordinal data types – potentially represented in the same systematic fashion as numbers. A particular challenge here is a data scarcity – for instance, as there are only seven days in a week, to warrant held-out evaluation, we can only fit the probes on six examples.

Prior work by Engels et al. (2025) demonstrates that LLMs internally represent temporal data – such as days of the week and months of the year in circular structures. Analytically, a centered circle is parameterized by the equations $x(\theta) = r\sin(\theta + \pi/2)$ and $y(\theta) = r\sin(\theta)$, indicating that these temporal concepts are encoded using sinusoidal features. Assuming this structure, we fit our probe on the embedding representations of these concepts and compare it to a prior probe architecture of Kadlčík et al. (2025) in leave-one-out cross-validation setting. The best held-out validation accuracies of both architectures are reported in Table 2. Our results demonstrate high decoding accuracy for ordinal data across all evaluated models. This validates the probe's utility beyond pure numbers, even under extreme data-scarcity. With our better generalizing architecture, we corroborate and extend the discovery by Engels et al. (2025) of circularity in LLMs' representations to other ordinal data types.

Model	sin	param-sin
Days of week (n=7)		
Olmo 2 1B	4	**6**
Olmo 2 7B	6	6
Olmo 2 13B	5	**6**
Llama 3 1B	5	**6**
Llama 3 3B	**7**	6
Llama 3 8B	3	**6**
Months (n=12)		
Olmo 2 1B	6	**11**
Olmo 2 7B	6	**11**
Olmo 2 13B	5	**11**
Llama 3 1B	9	**11**
Llama 3 3B	10	**11**
Llama 3 8B	6	**11**
Numerals: 'one' to 'ten' (n=10)		
Olmo 2 1B	3	**6**
Olmo 2 7B	4	**8**
Olmo 2 13B	4	**8**
Llama 3 1B	3	**5**
Llama 3 3B	4	**6**
Llama 3 8B	3	**9**

Table 2. Number of matches achieved by training param-sin and best-performing prior probe (sin) in decoding ordinal data with extremely small training sets from models' internals. Held-one-out evaluations.

B. Experimental details

The overview provided here is intended for informative purposes. For precise replication, we refer the reader to the companion codebase, which contains the exact hyperparameters for each experiment, available at `https://github.com/prompteus/numllama`

In our probe architecture *param-sin*, we initialize amplitude parameters $\mathbf{w}_a$ with a constant of 0.01, phase-shift parameters $\mathbf{w}_\phi$ randomly from uniform distribution between 0 and 2π, and frequency parameters $\mathbf{w}_f$ with a vector of values regularly-spaced between 0 and $\frac{\pi}{4}$, and the parameters of linear projection $\mathbf{W}_{\text{in}}$ randomly from uniform distribution between $-\sqrt{1/n}$ and $\sqrt{1/n}$, where d is the input dimension (number of model features).

We find that L1-regularization of the amplitude parameters $\mathbf{w}_a$ improves generalization to held-out tokens by encouraging the probe to maintain small effective latent dimensionality even with a large actual dimension d. Similarly, we found similar benefits from regularization of the linear projection $\mathbf{W}_{\text{in}}$, although L2-regularization appears more robust. We find that an analogous configuration is also effective for the probe of Kadlčík et al., with L2-regularization of $\mathbf{W}_{\text{in}}$ and L1-regularization of $\mathbf{W}_{\text{out}}$ for inducing simplicity

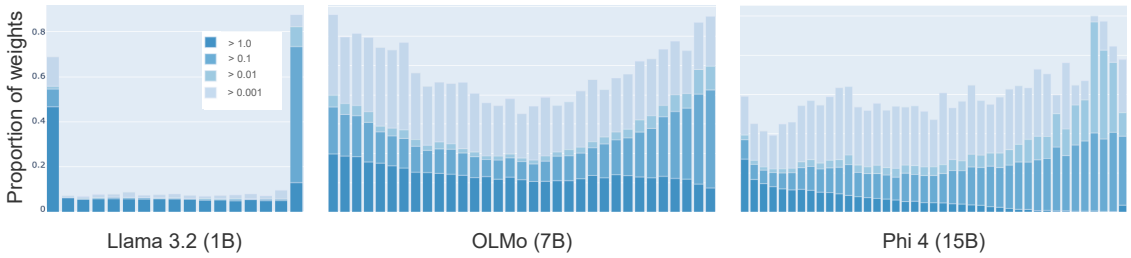


Figure 16. **Difference in probe weights distribution (of W_{in}) across layers** shows that the initial and last layers represent numbers in a systematically sparser fashion.

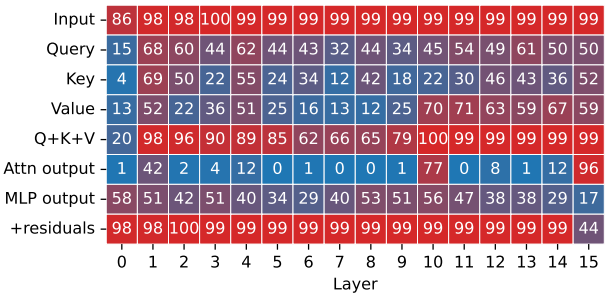
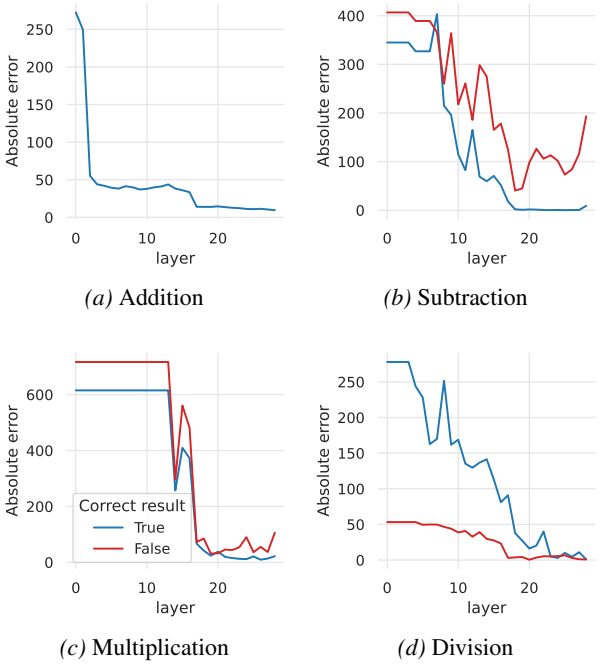


Figure 17. Accuracy of extracting input numbers from each component of the Transformer block using the sinusoidal probe.

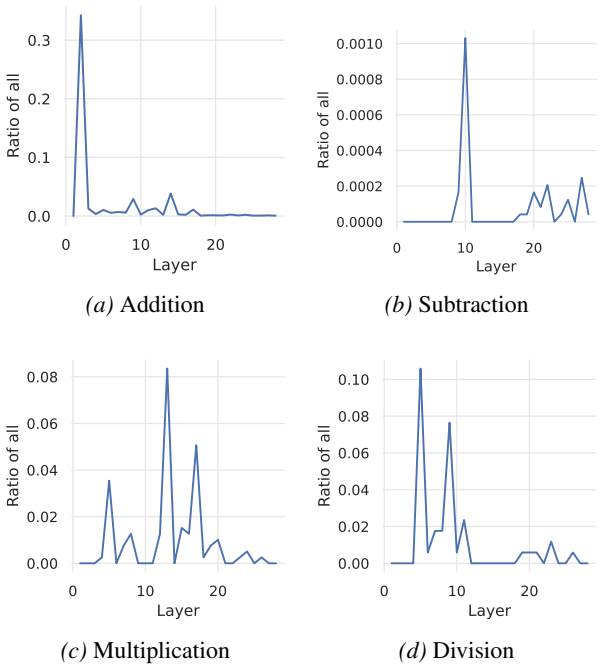


(a) Addition *(b)* Subtraction *(c)* Multiplication *(d)* Division

Figure 18. **Absolute error per layer:** Absolute error of numeric values probed from different layers.

(sparsity) in the sinusoidal basis.

The probes are fitted on a train set, with a checkpoint selected on the validation set, and then evaluated on the held-out test set. In most experiments, we use the Adam optimizer with a learning rate of 10^{-4} and apply regularization with a coefficient of either 10^{-3} or 10^{-2}.



(a) Addition *(b)* Subtraction *(c)* Multiplication *(d)* Division

Figure 19. **Error aggregation per layer:** Relative ratio of cases where each of the layers in Llama 3.2 3B *breaks* the correct result from the previous layer.

In multi-token decoding (Figure 6), we use a learning rate of $5 \cdot 10^{-4}$, 10,000 training steps, and early-stop training.

When fitting probes on natural language contexts (Figure 4, 'Nat. lang. $\rightarrow$ Nat. lang.') and for cross-layer transfer (Figure 12), we use a hidden dimension of $d = 500$.

In our steering experiments (Figure 9), for $f_{\sin}$, we optimize embeddings with respect to the probe using an SGD optimizer with a learning rate of 0.1, optimize the embeddings for 200,000 steps, and decrease the learning rate by a fac-

tor of 100 over the first 100,000 steps. As for $f_{\text{param sin}}$, we set regularization coefficient to 10^{-4} and use a hidden dimension of $d = 2000$.

We provide an overview of the datasets we use for natural language context experiments in Table 3. In Section 4.1, we use BBC data as processed by Li et al. (2024), which we retrieved from `RealTimeData/bbc_news_alltime`.

In all cases, we trust the original creators of the datasets for handling private and sensitive information according to standard. As we do not redistribute the data, we take no specific steps towards verifying this point.

C. Use of LLMs in writing and coding

LLM-based technology was used to polish writing and jump-start coding. Authors take full responsibility for the contents of this paper.

Domain	Dataset	Source	Numerical Context	Size
Culinary	Recipe NLG Lite	m3hrdadfi/recipe_nlg_lite	Quantities, measurements	6118
	FoodRecipe-ImageCaptioning	samsatp/FoodRecipe-ImageCaptioning	Ingredient amounts	719
Temporal	TimeLineExtraction	irlabamsterdam/TimeLineExtraction...CASE	Legal document dates	50
Arithmetic	MetaMathQA	meta-math/MetaMathQA	Mathematical reasoning	395K
	DROP	ucinlp/drop	Discrete reasoning	77.4K
	AQuA-RAT	deepmind/aqua_rat	Algebraic word problems	97.4K
Medical	ICD-10 Codes	atta00/icd10-codes	Diagnostic codes	25.7K
	ICD-10-CM	Gokul-waterlabs/ICD-10-CM	Medical classifications	74K

Table 3. Dataset specifications for numerical embedding analysis across natural language contexts.