AUTOREGRESSIVE MODELS REPRESENT MULTI-TOKEN NUMBERS AS PLACE-SCALED HELICES

Anonymous authors
Paper under double-blind review

ABSTRACT

Large numbers are often split across multiple tokens by traditional tokenizers in modern Language Models (LM), raising a fundamental question: how are the numerical values represented when no single token corresponds to the entire number? Prior work reads the whole number from one hidden state, usually the last token. We show that this last token correlates with the number but does not causally control the other digits, and therefore in this work we analyse each token's hidden state separately. We find that every token follows the same helical structure, with the period of the helix growing with the token's place. We call this the ***Place-Scaled Helix***. It correlates with the number representations better than prior structures, and intervening model activations using it changes the output digit, showing causal relevance. We have analysed the number representation across single-digit and multi-digit tokenizers, LLMs and VLMs, on arithmetic tasks showing a generalized numeric structure for autoregressive language models.

1 INTRODUCTION

Autoregressive models are now prevalent across a wide range of applications, and their ability to generate and manipulate numerical quantities has played an important role in advancing mathematical reasoning and numerical understanding. Recent works have shown that fine-tuning LMs on data containing numbers in their ground truth can teach models to represent and use numerical quantities directly, enabling improvements in downstream tasks without an explicit notion of the provenance, structure, or meaning of the numbers involved (Levy & Geva, 2025; Yuchi et al., 2026). Such numerical capabilities have also benefited domains beyond mathematical reasoning (Yue et al., 2023), including multimodal tasks such as counting (Deitke et al., 2024), visual question answering (Liu et al., 2023) and object detection (Kuckreja et al., 2023; Gupta et al., 2025). This was the motivation for us to find answers to questions such as: **1** How do the autoregressive models represent numbers internally? **2** Where in the model are the numerical representations formed? **3** How do these numerical representations causally affect the downstream task? Having an understanding of how numbers are represented can help us in strategizing the training paradigm better to improve numerical understanding, and in curating numerical datasets that align well with the internal structure of the LM. This is particularly important as current models can exhibit strong mathematical reasoning capabilities while still struggling with even basic numerical operations (Is $9.9 > 9.11$?) (Yang et al., 2025). Understanding the numerical representation could potentially bridge the gap between high-level mathematical reasoning and fundamental numerical understanding.

As shown in prior works (Levy & Geva, 2024; Kantamneni & Tegmark, 2025), aimed to understand how numbers are represented, but they limited their study to numbers represented by a single token, where the entire number is encoded within a single embedding. But in the real world, tokenizers split numbers across tokens; for instance, the multi-digit number 1251 can be split into 125 and 1, becoming multi-token, which breaks the notion of place and value, making it unclear how the model encodes it as a 4-digit number internally.

To recover the multi-token number representation, prior works used the last token embedding. We show that the final token can be correlated with the entire number, but this does not imply a causal role. When we causally intervene on the last token, it does not affect the numerical behavior. So, instead of analyzing the last token, we study each number token individually and find that each token represents the digit in the form of a helical representation consistent with (Kantamneni & Tegmark,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

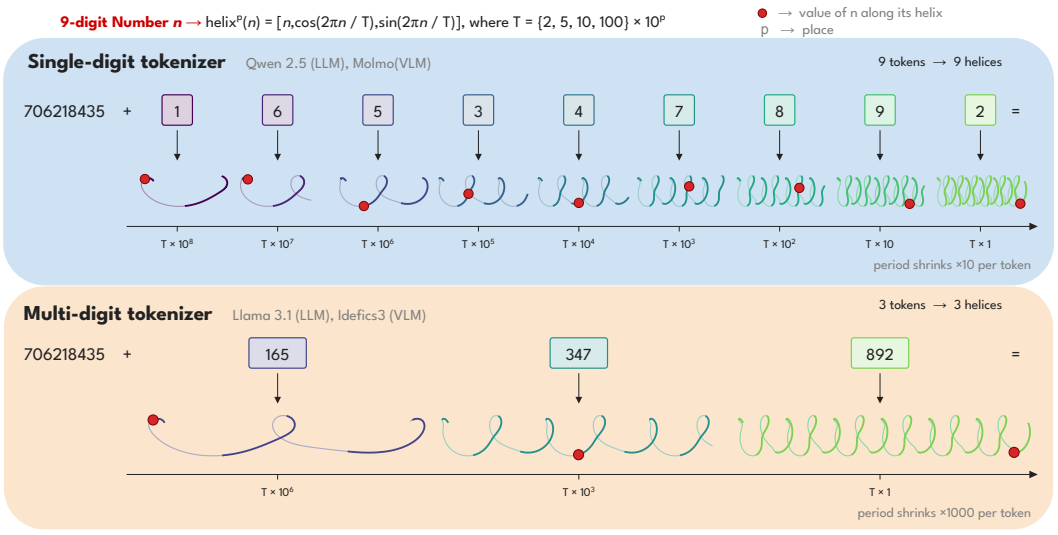

Figure 1: **Multi-token number representation.** We analysed the 9-digit operand $n = 165347892$ in the addition task of $706218435 + n$. The number representation inside the autoregressive LM, when the numbers are split across tokens, is represented by a helix at each token position. Further, the numeric period of the helix scales with token position starting from the units place (the last token has the highest frequency). Here, we show how a single-digit *(above)* and a multi-digit tokenizer *(below)* model represent a multi-token number.

Table 1: **Comparison of prior works on number representation.**

Previous Works	LLM	VLM	Multi-token	Token-wise geometry
Levy & Geva (2025)	✓	✗	✗	✗
Zhu et al. (2024)	✓	✗	✓	✗
Kajic & Nematzadeh (2022)	✗	✓	✗	✗
Kantamneni & Tegmark (2025)	✓	✗	✗	✗
Štefánik et al. (2026)	✓	✗	✓	✗
Ours	✓	✓	✓	✓

2025), but as we increase the place from units to tens to hundreds, the period of the helix structure scales with position; we call it the Place-Scaled Helix Fig. 1.

We further study when the structure is formed in the model and how it universally fits across LLMs/VLMs and across arithmetic tasks, showing a general structure of numerical representations rather than a model- or task-specific characteristic. Furthermore, we study its causal relevance by intervening on layer activations with that generated with the hypothesized Place-Scaled Helix. These interventions demonstrate that the discovered structure is not merely correlated with numerical values but plays a functional role in the model's downstream numerical behavior. Overall, our findings/contributions on the study of generalized numerical representations for autoregressive models are listed below:

- **Place-Scaled Numerical Representation.** *(Sec. 4)* We identify a common helical representation in LLMs/VLMs for multi-token numbers in which the periods at each numerical site scale systematically as 10^p with place p.
- **Causal Subspace Evaluation.** *(Sec. 5)* We evaluate numeric structure fits within a trained orthogonal low-dimensional subspace that isolates only numerical quantities. This allows us to evaluate and intervene to check the effect of the proposed structure only for numeric quantities.
- **Generalized Representation Study.** *(Sec. 6)* We perform comprehensive study on both single-digit and multi-digit tokenizers across LLMs and VLMs using a single place-indexed formulation, and evaluate the resulting structure across arithmetic tasks as shown in Tab. 1.

2 Related Work

Geometric representations of numbers. (Nanda et al., 2023) first observed circular Fourier representations in models trained on modular arithmetic, and (Zhou et al., 2024) found the same Fourier features associated with addition in pretrained LLMs. (Kantamneni & Tegmark, 2025) unified linear magnitude and periodic structure through a generalized helix, with periods $T \in \{2, 5, 10, 100\}$, and showed that the model uses this helix for addition through the Clock algorithm. (Levy & Geva, 2025) identified base-10 circular representations for individual digits, and (Kadlčík et al., 2025) showed that sinusoidal probes can recover numerical values with high accuracy. These works establish linear and periodic geometry as recurring properties of numerical representations, but they fit it in the full residual stream, where number geometry is mixed with text, operators and, for VLMs, images. We fit within a subspace that isolates numeric content.

Multi-token numerical representations. (Zhu et al., 2024) show that the value of a number can be decoded progressively as its tokens are processed, including from intermediate positions, but they characterize the scalar value rather than the geometry at each position. (Levy & Geva, 2025) also analyze digit-tokenized numbers, but recover digit information from the final numeric token, and their causal intervention is run on numbers 0–999, which Llama 3 encodes as a single token. Most closely related, (Štefánik et al., 2026) show that preceding pieces of a multi-token number remain recoverable from its final-token representation. This shows that numerical information is retained in a later hidden state, but not that the model reads the number from there, and none of these works test whether intervening on the final token changes the other places. We characterize the hidden state at each numerical site and find that the same helical form holds at every position, with its period scaling with decimal place.

Numbers in Vision-Language Models. Prior multimodal work probes how VLMs represent counts read from an image (Kajic & Nematzadeh, 2022; Pang et al., 2026; Hasani et al., 2026), but does not characterize the geometry of multi-token numbers that the model generates. We find the same place-scaled helix in both LLMs and VLMs.

3 Generalised Number Representation

In this section, we analyze the characteristics of the number representation for single-token numbers and transition to the structure for multi-token number representation.

3.1 Setup

We study both the Single-digit Tokenizer: ***Molmo2-8B (VLM)***, ***Qwen2.5-7B-Instruct***, and Multi-digit Tokenizer: ***Idefics3-8B (VLM)***, ***Llama3.1-8B***, which have unique embeddings for the numbers [0–9] and [0–999], respectively. We focus only on representing natural numbers, following prior work (Kantamneni & Tegmark, 2025), but we experiment with multi-token numbers. We conduct most of our analysis on the addition task with 9-digit operands and answers (unless otherwise specified), which corresponds to 9 tokens for single-digit tokenizers and 3 tokens for multi-digit tokenizers. We discuss the characteristics of number representation for multi-token numbers in the following section. The analysis is performed on Molmo2 and Llama3.1. Further results on other models can be found in *Supplementary*.

3.2 Single-Token Number Representation

Monotonicity. We apply Singular Value Decomposition (SVD) on top of the layer activations to get the major Principal Components (PC) for the token corresponding to the number. We notice that contiguous clusters are formed within the *PC1–PC2* plane without any non-linear dimensionality reduction like t-SNE or UMAP (McInnes et al., 2020), as shown in Fig. 2 (*left*). It shows that the number representations are monotonic and are approximately linearly separable along a PC1 projection measured by

$$\mathrm{PC1}_{\mathrm{score}}{}^{(l)} = (h_l - \mu_l)^\top v_{\mathrm{PC1}}^{(l)},$$

where h_l is the residual stream, μ_l is the sample mean, and $v_{\mathrm{PC1}}^{(l)}$ is the *PC1* vector at layer l.

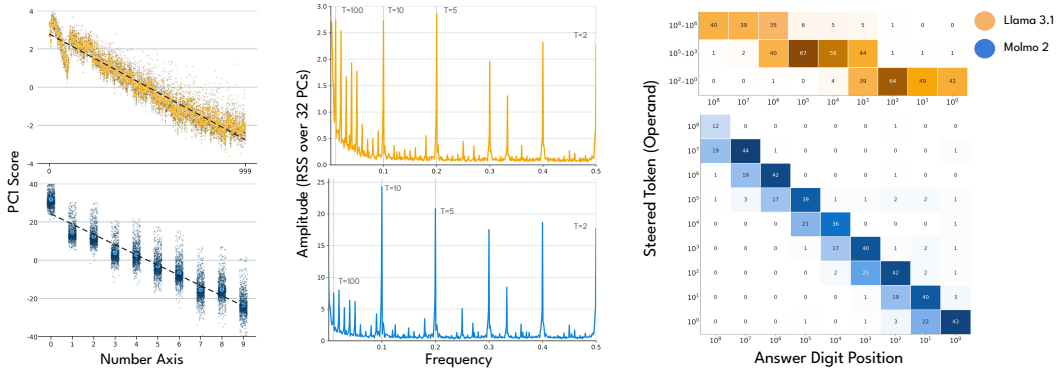

Figure 2: **Properties of Number Representation.** *(Left)* **Linearity & Monotonicity.** The PC1 score changes linearly with the number and the structure is contiguous, making it monotonic. *(Mid)* **Periodicity.** Fourier decomposition reveals periodic structure in the numerical representations. *(Right)* **Steerability.** Multi-digit and single-digit tokenizer models can change only the digits corresponding to the steered token, and one token to the left because of carry-over. This suggests that the steering vector varies with the place, which says there exists a different structure for each place.

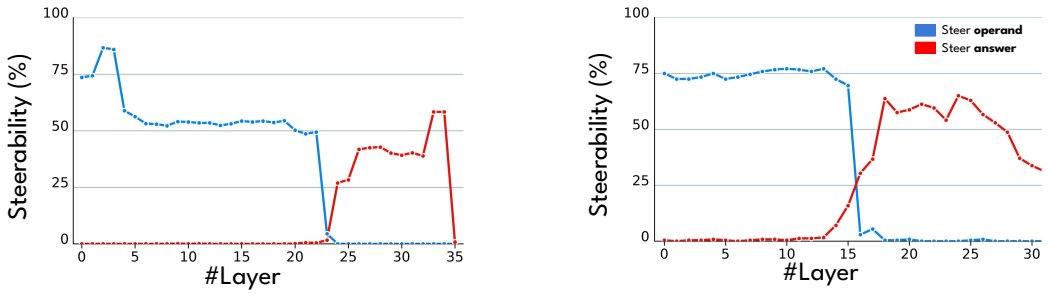

Figure 3: **Steerability with input vs. output.** We compute the % change in the output answer when we steer the operand (input) and when we steer the answer (output). Earlier layers show steerability with input and later layers with output. Here *(Left)* is for Molmo2 and *(Right)* is for Llama3.1.

Linearity. In Fig. 2 (*left*), we can see that, as we scale the numbers, the absolute value of PC1 increases steadily, proving that linearity exists.

Periodicity. We apply Fourier decomposition of the continuous number domain over the Root-Sum-Squared (RSS) of the top 32 PCs to the frequency domain. We notice that the number periods are $T \in \{2, 5, 10, 100\}$, as shown in Fig. 2 (*mid*).

Generally, the number representation always has monotonicity, periodicity, and linearity terms for a single-token number representation. So, following (Kantamneni & Tegmark, 2025), we assume that the structure is helical for a number y:

$$\varphi(y) = \left[y,\ \cos\tfrac{2\pi y}{T},\ \sin\tfrac{2\pi y}{T}\right]_{T\in\mathcal{T}}, \quad \mathcal{T} = \{T_0 \mid T_0 \in \{2, 5, 10, 100\}\}, \tag{1}$$

where

$$y \in \begin{cases} \{0, \ldots, 9\}, & \text{single-digit tokenizer,} \\ \{0, \ldots, 999\}, & \text{multi-digit tokenizer.} \end{cases}$$

All prior works base their analysis on having unique embeddings for numbers; we now check how the representation looks when numbers are split into multiple tokens.

3.3 Multi-Token Number Representation

For multi-token numbers, the value of the number is spread across positions, which raises questions such as: does each token hold only its own digits? Does the model combine the tokens into one quantity? A linear probe cannot answer this, because it only shows that a token's residual *correlates* with its digits, but cannot answer whether the model *uses* them. So we intervene directly. In the task of 9-digit operand addition $a + b = c$, we steer operand a in the initial layers and check which digits of the output change, as shown in Fig. 3.

Causal Steering over Linear Probing. At token k and layer l, we fit a ridge probe from the residual h_k to that token's own digit(s), take its unit direction w, and steer,

$$h_k \leftarrow h_k + \alpha\, \sigma\, w, \tag{2}$$

where σ is the standard deviation of the data projected onto w, so the scalar α is measured in units of the model's own spread. The embedding of the last token contains the entire number, which can be validated with linear probing, as discussed in (Štefánik et al., 2026). Yet causal interventions on that last token do not alter other places, as shown in Fig. 2 (*right*). This suggests correlation does not imply causation.

Occurrence of Place and Value structure. Each place now holds a different ridge vector for causal steering, showing a different structure for each place. We analyzed output c on the major PC plane across layers in Fig. 4; it shows that clustering for place occurs much earlier in the model and stays clustered throughout, while the value cluster is formed much later. This remains our main motive for a fit which is indexed based on place, as early formation of place-specific clusters suggests that each token position develops a distinct representational structure. Within this structure, the value exhibits a monotonic or near monotonic trend, which suggests that the fit will become causally relevant only in the later layers, through formation of the helical structure. Importantly, within each place cluster, the variance ρ_L stays near 0.39 up to *layer 22*, then doubles to 0.78 by *layer 30*. This increase in variance indicates that the structure progressively expands as the representations propagate to deeper layers. More details are provided in the *Supplementary Material*.

3.4 Numerical Structure is Modality Agnostic

To understand the numeric structure for VLM when we provide input as images, we render a single number in an image and ask the model to read it. Molmo2 reads 9-digit numbers (99.6% accuracy) but Idefics3 cannot (0%), so it reads 3-digit numbers (96%). Tab. 2 shows that the DAS subspace trained on *text* fits the image representation almost equally well without retraining (Molmo2 - 0.48 vs. 0.50 and Idefics3 - 0.30 vs. 0.29), indicating that numbers read from images share the representation used for text and the same Place-Scaled Helix structure fits better.

4 Place-Scaled Helix Structure

Single-token numbers in LLMs are represented as a normal helix, which has a linear component encoding magnitude combined with sinusoidal components at periods $T \in \{2, 5, 10, 100\}$ (Kantamneni & Tegmark, 2025). But it fails to generalize for multi-token numbers Tab. 3. We further identified three constraints a multi-token numeric structure must satisfy:

1 Each token should have linearity and periodicity (helix form); **2** The basis must be indexed by place, since each place holds different structure; **3** Expansion of structure with place, suggested by increase in variance per cluster in Fig. 4.

The minimal basis satisfying all three is a helix whose periods scale with place as represented in Fig. 1. For a number y at site p:

$$\varphi_p(y) = \left[\, y,\ \cos\tfrac{2\pi y}{T},\ \sin\tfrac{2\pi y}{T} \,\right]_{T \in \mathcal{T}_p}, \quad \mathcal{T}_p = \{T_0 \cdot 10^{\alpha p} \mid T_0 \in \{2, 5, 10, 100\}\}, \tag{3}$$

where $\alpha = 1$ for single-digit tokenizers and $\alpha = 3$ for multi-digit tokenizers, since at place p, a multi-digit token spans three orders of magnitude as it represents $[0, 999]$ in one token.

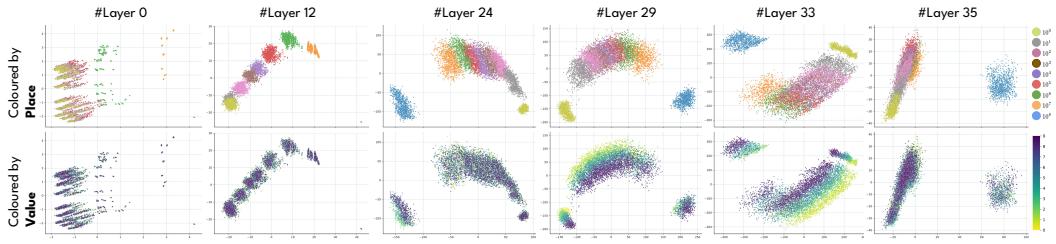

Figure 4: **Clustering of Place and Value.** The place is encoded in distinctive clusters much earlier compared to value, suggesting a different representation per place. The value formation within each place cluster suggests the formation of number representation structure. Analyzed on the answer tokens of Molmo2. More details in *Supplementary*.

		Molmo2-8B (L34) (9-digit number)																	Idefics3-8B (L20), (3-digit number)	
Token →		Image Trained DAS-20									Text Trained DAS-20 (transfer)								Image Trained DAS-20	Text Trained DAS-20 (Transfer)
Fit ↓		10^8	10^7	10^6	10^5	10^4	10^3	10^2	10^1	10^0	10^8	10^7	10^6	10^5	10^4	10^3	10^2	10^1	10^0	10^2–10^0	10^2–10^0
Place-Scaled Helix	10^8	**.77**	.05	-.01	-.01	.00	-.01	-.01	-.01	-.01	**.71**	.03	-.01	-.01	.00	-.01	-.01	-.01	-.01	–	–
	10^7	.46	**.48**	.00	-.01	-.01	-.01	-.01	-.01	-.01	.32	**.45**	-.01	-.01	-.01	-.01	-.01	-.01	-.01	–	–
	10^6	.18	.20	**.46**	-.01	-.01	-.01	-.01	-.01	-.01	.11	.17	**.44**	-.01	-.01	-.01	-.01	-.02	-.01	–	–
	10^5	.19	.00	.20	**.44**	.00	-.01	-.01	-.01	-.01	.11	-.01	.17	**.41**	-.01	-.01	-.01	-.01	-.01	–	–
	10^4	.19	.00	-.01	.20	**.46**	.00	-.01	-.01	-.01	.11	-.01	-.01	.16	**.42**	-.01	-.01	-.01	-.01	–	–
	10^3	.18	-.01	-.01	.00	.20	**.45**	-.01	-.01	-.01	.11	-.01	-.02	-.01	.17	**.42**	-.01	-.01	-.01	–	–
	10^2	.18	-.01	-.01	-.01	.00	.19	**.46**	-.01	-.01	.10	-.01	-.01	-.01	-.01	.16	**.43**	-.01	-.01	**.28**	**.30**
	10^1	.19	-.01	-.02	-.01	-.01	-.01	.19	**.47**	-.01	.11	-.01	-.02	-.01	-.01	-.01	.16	**.44**	-.01	**.30**	**.32**
	10^0	.19	.00	-.01	-.01	-.01	-.01	-.01	.20	**.55**	.11	-.01	-.01	-.01	-.01	-.01	-.01	.16	**.56**	**.30**	**.30**
Other Fits	place-scaled circle	.45	.21	.19	.18	.19	.18	.19	.19	.22	.32	.17	.17	.15	.16	.16	.16	.16	.21	.12	.12
	normal helix	.19	.00	-.01	-.01	-.01	-.01	-.01	.20	.55	.11	-.01	-.01	-.01	-.01	-.01	-.01	.16	.56	.30	.30
	circle	.19	.00	.00	-.01	-.01	-.01	.00	.00	.22	.12	.00	.00	-.01	-.01	-.01	-.01	.00	.21	.12	.12
	log(y)	.26	.02	.00	.00	.00	.00	.00	.00	.00	.14	.01	.00	.00	.00	.00	.00	.00	.00	.05	.04
	linear(y)	.19	.01	.00	.00	.00	.00	.00	.00	.00	.12	.00	.00	.00	.00	.00	.00	.00	.00	.05	.04

Table 2: R^2 **for Number Representation Derived from Image**. Molmo2 reads a 9-digit number (one token per digit) and Idefics3, which cannot read 9-digit numbers from an image, reads a 3-digit one (one token). *(Image Trained DAS)* For each answer token we train a 20-dim DAS subspace on this image task and fit the token's image activations inside it. It shows R^2 is maximum for Place-Scaled helix following the text only structure. *(Text Trained DAS)* The same image activations fitted inside the 20-dim DAS subspace trained on text addition, with no retraining. The fit is almost unchanged (Molmo2 - 0.48 vs 0.50; Idefics3 - 0.30 vs 0.29), so the number read from an image lives in the subspace learned from text.

For n samples, the design matrix $\mathbf{\Phi}_p \in \mathbb{R}^{n\times m}$ stacks $\boldsymbol{\varphi}_p(y_i)^\top$ row-wise, with $m = 2|\mathcal{T}_p| + 1$. We validate this choice against the unscaled helix (Kantamneni & Tegmark, 2025), circle, logarithmic, and linear fits, and find that each site's representation is best described by the place-scaled helix when scaled by 10 as shown in Fig. 5 *(left)*.

5 Evaluation of Numeric Structure using Aligned Subspace

The residual stream at layer ℓ, $\mathbf{h}^\ell \in \mathbb{R}^d$, carries the variance in the input prompt, which includes text, operations, and other modalities (i.e., images), not only the numbers. Fitting the helix directly in $\mathbb{R}^d$ would lead to improper formulation of numeric structure with these other sources of variance. We therefore first isolate a low-dimensional subspace that captures numeric content specifically.

Using Distributed Alignment Search (DAS) (Geiger et al., 2024), we learn k orthonormal directions $\mathbf{R}^\ell \in \mathbb{R}^{d\times k}$ for each layer and answer position token pair, giving the projected code $\mathbf{c}^\ell = \mathbf{R}^{\ell\top}\mathbf{h}^\ell \in \mathbb{R}^k$. We then fit the place-scaled basis within this subspace via ridge regression:

$$\mathbf{W}^* = \left(\mathbf{\Phi}_p^\top \mathbf{\Phi}_p + \lambda \mathbf{I}\right)^{-1} \mathbf{\Phi}_p^\top \mathbf{C}, \tag{4}$$

where $\mathbf{C} \in \mathbb{R}^{n\times k}$ stacks the projected codes. Fit quality is measured by the mean coefficient of determination $\bar{R}^2 = \frac{1}{k}\sum_j R_j^2$ under 5-fold cross-validation. To confirm that the model uses this subspace, we perform activation patching (Heimersheim & Nanda, 2024) restricted to its k dimensions. Let $\mathbf{P} = \mathbf{R}\mathbf{R}^\top$ be the orthogonal projector onto the sub-

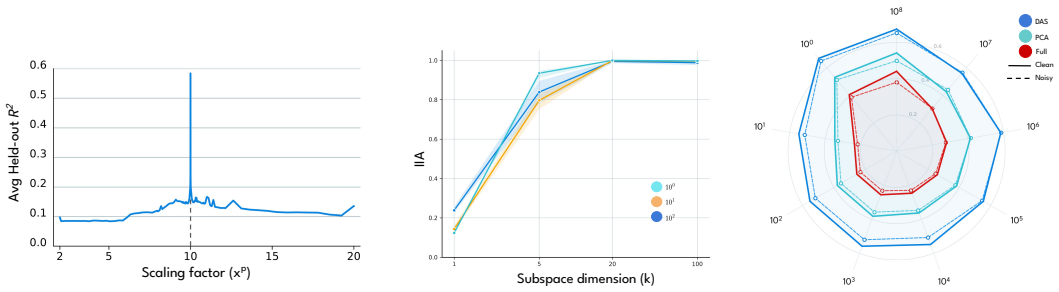

Figure 5: *(Left)* **Scaling Factor in Place-Scaled Helix.** The base 10^p achieves the highest R^2 when averaged across all token positions, potentially reflecting the model's base-10 numerical structure. *(Mid)* **Finding Numeric Subspace.** The $k = 20$ trained orthonormal vectors are sufficient to represent any number, achieving 100% intervention accuracy. *(Right)* **Superiority of Aligned Subspace.** The R^2 is highest with the $k = 20$ subspace trained through DAS, exceeding the fit obtained using the full 4096-dimensional space. Tested on 9-token number representations on noisy data.

space spanned by the columns of $\mathbf{R} \in \mathbb{R}^{d\times k}$. Given a target number y', we write the fitted code $\hat{\mathbf{c}} = \mathbf{W}^{*\top}\boldsymbol{\varphi}_p(y')$ into the subspace and leave its complement untouched:

$$\mathbf{h}_{\text{new}} = (\mathbf{I}_d - \mathbf{P})\,\mathbf{h} + \mathbf{R}\,\hat{\mathbf{c}}. \tag{5}$$

We patch only the k numeric-aligned coordinates, and the component of $\mathbf{h}$ in the remaining $d - k$ dimensions is preserved exactly, as shown in Fig. 6. If $\hat{\mathbf{c}}$ is the code the model reads, the patched forward pass should emit the digit of y' at that position. We measure this as interchange intervention accuracy (IIA), which is the fraction of sums for which the patched output equals the digit of y'. Prior work (Kantamneni & Tegmark, 2025) focused on logit difference, but accuracy clearly shows the causal effect as it only measures if the output digit changes after interventions, not just the logits.

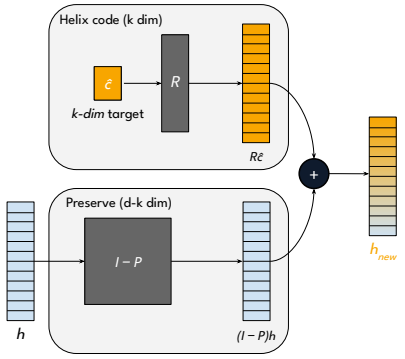

Figure 6: k**-dim Activation Patching.** We patch only the number-aligned k-dim, which contains the numeric representation from a fit to check for downstream causal intervention accuracy.

Complete causal intervention is possible using only the *20-dim* subspace of *4096-dim* as shown in Fig. 5 (*mid*), which shows that we can evaluate the causal effect of the fit better in a subspace where only numbers are represented. Patching with the *20-dim* aligned subspace on the *4096-dim* residual stream yields a better fit than patching with the top-20 principal components or the full dimension, at every answer position on noisy data Fig. 5 (*right*). The aligned subspace maintains R^2 consistently above 0.5 even when the noisy set contains irrelevant text and images prior to the addition task. Addition of noisy images, still doesn't affect the numeric structure, supporting the claims of numeric structure being modality agnostic. More details in *Supplementary*.

6 Results

In this section, we address three key questions: which layers matter, what constitutes a good fit for number representation, and whether the quality of the fit is causally relevant. We provide a quantitative analysis across tasks and models to systematically evaluate these questions.

Which layer shows the most causal relevance after patching?. To evaluate the fit we need to choose the layers where the fit has the maximum causal effect. To study at which layer the fit is formed, we show R^2 across all places for different fits in Fig. 7. The Helix fits the model representations from Layer *26* and saturates, this can be viewed as formation of value around 26-34

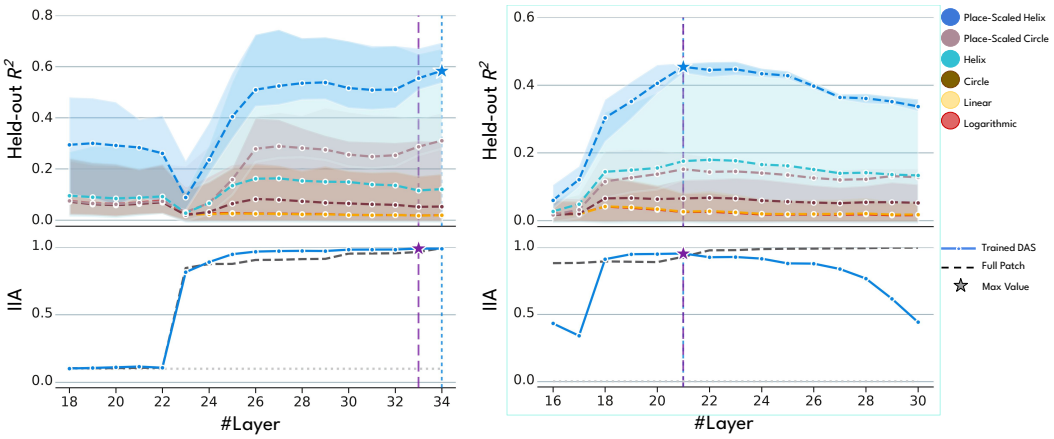

Figure 7: **Emergence of Causality.** Place-Scaled Helix shows better fit compared to other structures and reaches maximum for (*Left*) Molmo2 at layer 32-34 and (*Right*) Llama3.1 at layer 22-24.

Token → / Fit ↓	10^8	10^7	10^6	10^5	10^4	10^3	10^2	10^1	10^0	10^8	10^7	10^6	10^5	10^4	10^3	10^2	10^1	10^0	10^8	10^7	10^6	10^5	10^4	10^3	10^2	10^1	10^0	10^8	10^7	10^6	10^5	10^4	10^3	10^2	10^1	10^0
Molmo2-8B (L34)																																				
Place-Scaled Helix 10^8	**.69**	.01	-.01	-.01	-.01	-.01	-.01	-.01	-.01	**.67**	.01	-.01	-.01	-.01	-.01	-.01	-.01	.00	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Place-Scaled Helix 10^7	.43	**.56**	-.01	-.01	-.01	-.01	-.02	-.01	-.01	.39	**.59**	-.01	-.01	-.01	-.01	-.01	-.01	-.01	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Place-Scaled Helix 10^6	.18	.29	**.58**	-.01	-.01	-.01	-.02	-.01	-.02	.13	.32	**.58**	-.01	-.01	-.01	-.02	-.01	-.01	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Place-Scaled Helix 10^5	.17	.00	.32	**.56**	-.01	-.01	-.01	-.01	-.01	.12	.00	.33	**.56**	-.01	-.01	-.01	-.01	-.01	–	–	–	**.66**	.00	.00	-.01	-.01	-.01	–	–	–	**.67**	.00	-.01	-.01	-.01	-.01
Place-Scaled Helix 10^4	.17	-.01	.00	.31	**.55**	-.01	-.01	-.01	-.01	.12	-.01	.00	.33	**.58**	.00	-.01	-.01	-.01	–	–	–	.33	**.48**	.00	-.01	-.01	-.01	–	–	–	.33	**.46**	.00	-.01	-.02	-.01
Place-Scaled Helix 10^3	.17	-.01	-.01	.00	.30	**.56**	-.01	-.01	-.01	.12	-.01	-.01	.01	.34	**.58**	-.01	-.01	-.01	–	–	–	.12	.22	**.46**	-.01	-.01	-.01	–	–	–	.13	.23	**.43**	-.01	-.01	-.01
Place-Scaled Helix 10^2	.17	-.01	-.01	-.01	.00	.31	**.55**	-.01	-.01	.12	-.01	-.01	-.01	.00	.34	**.56**	-.01	-.02	–	–	–	.12	-.01	.23	**.42**	.00	-.01	–	–	–	.13	-.01	.23	**.33**	-.01	-.01
Place-Scaled Helix 10^1	.17	-.01	-.01	-.01	-.01	.00	.30	**.54**	.00	.12	-.01	-.01	-.01	-.01	-.01	.32	**.54**	.00	–	–	–	.12	-.01	-.01	.20	**.47**	-.01	–	–	–	.13	-.01	-.01	.16	**.34**	-.01
Place-Scaled Helix 10^0	.17	-.01	-.01	-.01	-.01	-.01	.30	**.67**	.13	-.01	-.01	-.01	-.01	-.01	-.01	.31	**.66**	–	–	–	.12	-.01	.00	.00	.21	**.57**	–	–	–	.12	-.01	-.01	-.01	.17	**.37**
Other Fits: place-scaled circle	.42	.29	.31	.30	.30	.30	.30	.28	.31	.37	.31	.32	.31	.33	.33	.32	.30	.32	–	–	–	.33	.23	.24	.20	.20	.21	–	–	–	.33	.23	.24	.17	.17	.17
Other Fits: normal helix	.17	-.01	-.01	-.01	-.01	-.01	-.01	.30	**.67**	.13	-.01	-.01	-.01	-.01	-.01	-.01	.31	**.66**	–	–	–	.12	-.01	.00	.00	.21	**.57**	–	–	–	.12	-.01	-.01	-.01	.17	.37
Other Fits: circle	.18	.00	.00	.00	.00	.00	-.01	.01	.31	.13	.00	.00	.00	.00	-.01	-.01	.01	.32	–	–	–	.13	.00	.00	.00	.00	.21	–	–	–	.13	.00	-.01	-.01	-.01	.17
Other Fits: log(y)	.18	.00	.00	.00	.00	.00	.00	.00	.00	.14	.00	.00	.00	.00	.00	.00	.00	.00	–	–	–	.13	.00	.00	.00	.00	.00	–	–	–	.13	.00	.00	.00	.00	.00
Other Fits: linear(y)	.18	.00	.00	.00	.00	.00	.00	.00	.00	.13	.00	.00	.00	.00	.00	.00	.00	.00	–	–	–	.13	.00	.00	.00	.00	.00	–	–	–	.13	.00	.00	.00	.00	.00
Qwen2.5-7B-Instruct (L26)																																				
Place-Scaled Helix 10^8	**.68**	.02	-.01	-.01	-.01	-.01	-.01	.00	-.01	**.71**	.02	.00	-.01	.00	-.01	.00	-.01	.00	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Place-Scaled Helix 10^7	.46	**.53**	-.01	-.01	-.01	-.01	-.01	-.01	-.01	.46	**.55**	.00	-.01	-.01	-.01	-.01	-.01	-.01	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Place-Scaled Helix 10^6	.21	.32	**.55**	-.01	-.01	-.01	-.01	-.01	-.01	.21	.31	**.54**	-.01	-.01	-.01	-.01	-.01	-.01	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
Place-Scaled Helix 10^5	.16	.06	.29	**.51**	-.01	-.01	-.01	-.01	-.01	.19	.03	.28	**.50**	.00	-.01	-.01	-.01	-.01	–	–	–	**.62**	.03	.00	-.01	-.01	-.01	–	–	–	**.65**	.03	.00	-.01	-.01	-.01
Place-Scaled Helix 10^4	.16	-.01	.04	.31	**.48**	.00	-.01	-.01	-.01	.19	-.01	.04	.30	**.52**	.00	-.01	-.01	-.01	–	–	–	.34	**.42**	.00	-.01	-.01	-.01	–	–	–	.35	**.35**	.00	-.01	-.01	-.01
Place-Scaled Helix 10^3	.15	-.01	-.01	.06	.28	**.49**	-.01	-.01	-.01	.19	-.01	-.01	.06	.31	**.54**	.00	-.01	-.01	–	–	–	.15	.22	**.27**	-.01	-.01	-.01	–	–	–	.14	.18	**.17**	.00	-.01	-.01
Place-Scaled Helix 10^2	.16	-.01	-.01	-.01	.06	.26	**.49**	-.01	-.01	.19	-.01	-.01	-.01	.04	.28	**.55**	-.01	-.01	–	–	–	.14	.02	.12	**.28**	-.01	-.01	–	–	–	.13	.01	.07	**.18**	.00	-.01
Place-Scaled Helix 10^1	.16	-.01	-.01	-.01	-.01	.03	.25	**.51**	-.01	.19	-.01	-.01	-.01	-.01	.02	.28	**.53**	.00	–	–	–	.14	.02	.00	.11	**.30**	.00	–	–	–	.14	.01	-.01	.06	**.17**	.00
Place-Scaled Helix 10^0	.16	-.01	-.01	-.01	-.01	-.01	.03	.26	**.55**	.19	-.01	.00	-.01	.00	-.01	.02	.26	**.60**	–	–	–	.15	.02	.01	.02	.13	**.50**	–	–	–	.14	.01	.00	.00	.07	**.21**
Other Fits: place-scaled circle	.40	.24	.24	.23	.20	.22	.20	.22	.22	.43	.27	.23	.22	.25	.24	.25	.23	.23	–	–	–	.34	.23	.12	.11	.10	.17	–	–	–	.34	.19	.07	.07	.06	.07
Other Fits: normal helix	.16	-.01	-.01	-.01	-.01	-.01	.03	.26	.55	.19	-.01	.00	-.01	.00	-.01	.02	.26	**.60**	–	–	–	.15	.02	.01	.02	.13	**.50**	–	–	–	.14	.01	.00	.00	.07	.21
Other Fits: circle	.16	.00	.00	.00	-.01	-.01	.00	.03	.22	.20	.00	.00	.00	.00	-.01	.00	.03	.23	–	–	–	.15	.02	.00	.00	.00	.17	–	–	–	.14	.02	.00	.00	.00	.07
Other Fits: log(y)	.15	.00	.00	.00	.00	.00	.00	.00	.00	.19	.00	.00	.00	.00	.00	.00	.00	.00	–	–	–	.14	.03	.00	.00	.00	.00	–	–	–	.13	.02	.00	.00	.00	.00
Other Fits: linear(y)	.16	.00	.00	.00	.00	.00	.00	.00	.00	.20	.00	.00	.00	.00	.00	.00	.00	.00	–	–	–	.15	.02	.00	.00	.00	.00	–	–	–	.14	.02	.00	.00	.00	.00

Table 3: **5-fold CV R^2 fit on single-digit tokenizer.** Each column is an answer token (the place it emits) across 4 tasks. Each row fits that token's 20-dim subspace with the place-scaled helix built for place 10^p (the Place-Scaled Helix rows).

Layer in Fig. 4. We show IIA at every layer to choose which layer the fit is causally active, and we choose Layer 34 for Molmo2 to be the best layer. More details in *Supplementary.*

What is a good fit for number representation? After choosing the best layers, we evaluate the Fit on single-digit tokenizer Tab. 3 and multi-digit tokenizer Tab. 4 on 4 arithmetic tasks, addition, subtraction, multiplication and division. For addition/subtraction we used the standard 9 digit numbers, whereas for multiplication/division, due to poor accuracy, we use 6 digit numbers instead. This shows place-scaled helix fits closely to arithmetic tasks over multi-token numbers across models. Since, multi-digit tokenizers encodes a range of [0-999], we can intervene on 3 places with one token. Additionally, we added place-scaled circle fit, where the periods of cos and sin are scaled by 10^p. This shows improvement over normal circle, suggesting scaling period is relevant.

How causally relevant is the quality of the fit? R^2 metric would give us the measure of a good fit, but to check if model uses this fit, we show causal intervention accuracy by applying this fit. As shown in Tab. 5, we compared fits across tokenizers and tasks, this shows the IIA averaged across places is the maximum for place-scaled helix. However, the multiplication and division shows poor fit for multi-digit tokenizer and poor causal effects on both tokenizers, this shows additional structure exists for these task.

Token → / Fit ↓		+ 10^8–10^6	+ 10^5–10^3	+ 10^2–10^0	− 10^8–10^6	− 10^5–10^3	− 10^2–10^0	× 10^8–10^6	× 10^5–10^3	× 10^2–10^0	÷ 10^8–10^6	÷ 10^5–10^3	÷ 10^2–10^0
Llama-3.1-8B (L24)													
Place-Scaled Helix	–	**.28**	.00	-.01	**.28**	.00	-.01	–	–	–	–	–	–
	–	**.43**	-.01	-.01	**.41**	.00	-.01	–	–	–	–	–	–
	10^6	**.42**	-.01	-.01	**.42**	.00	-.01	–	–	–	–	–	–
	–	.16	**.14**	-.01	.18	**.14**	-.01	–	**.18**	-.01	–	**.17**	.01
	–	.05	**.37**	-.01	.06	**.37**	-.01	–	**.28**	-.01	–	**.25**	.01
	10^3	.05	**.43**	-.01	.06	**.40**	-.01	–	**.17**	.00	–	**.11**	.03
	–	.05	.13	**.10**	.06	.11	**.13**	–	.07	**.04**	–	.04	**.05**
	–	.05	-.01	**.42**	.06	-.01	**.46**	–	.04	**.12**	–	.03	**.05**
	10^0	.06	-.01	**.45**	.06	-.01	**.43**	–	.06	**.29**	–	.03	**.07**
Other Fits	place-scaled circle	.17	.13	.12	.18	.12	.11	–	.07	.05	–	.04	.02
	normal helix	.06	-.01	.45	.06	-.01	.43	–	.06	.29	–	.03	.07
	circle	.06	-.01	.12	.07	.00	.11	–	.05	.05	–	.04	.02
	log(y)	.06	.00	.00	.07	.00	.00	–	.05	.00	–	.04	.01
	linear(y)	.06	.00	.00	.07	.00	.00	–	.05	.00	–	.04	.01
Idefics3-8B-Llama3 (L20)													
Place-Scaled Helix	–	**.29**	.00	-.01	**.33**	.00	.00	–	–	–	–	–	–
	–	**.35**	-.01	-.01	**.36**	.00	-.01	–	–	–	–	–	–
	10^6	**.51**	.00	-.01	**.44**	.00	-.01	–	–	–	–	–	–
	–	.22	**.15**	-.01	.19	**.15**	-.01	–	**.17**	.00	–	**.21**	.01
	–	.10	**.34**	.00	.08	**.35**	.00	–	**.09**	.00	–	**.09**	.00
	10^3	.09	**.43**	.00	.07	**.42**	.01	–	**.08**	.00	–	**.05**	.03
	–	.09	.11	**.16**	.07	.11	**.17**	–	.03	**.03**	–	.02	**.01**
	–	.09	-.01	**.44**	.08	.00	**.44**	–	.02	**.06**	–	.01	**.01**
	10^0	.09	-.01	**.38**	.08	-.01	**.41**	–	.04	**.29**	–	.02	**.03**
Other Fits	place-scaled circle	.22	.11	.10	.19	.11	.11	–	.04	.04	–	.02	.01
	normal helix	.09	-.01	.38	.08	-.01	.41	–	.04	.29	–	.02	.03
	circle	.10	.00	.10	.08	.00	.11	–	.03	.04	–	.02	.01
	log(y)	.09	.00	.00	.07	.00	.00	–	.02	.01	–	.02	.01
	linear(y)	.10	.00	.00	.08	.00	.00	–	.03	.00	–	.02	.01

Table 4: **5-fold CV R^2 fit on multi-digit tokenizer.** Each column is an answer token (the place it emits). Each row fits that token's 20-dim subspace with the place-scaled helix built for place 10^{3p}.

Fit	Single-digit tokenizer								Multi-digit tokenizer							
	Molmo2-8B (L34)				Qwen2.5-7B-Instruct (L26)				Llama-3.1-8B (L24)				Idefics3-8B-Llama3 (L20)			
	+	−	×	÷	+	−	×	÷	+	−	×	÷	+	−	×	÷
log(y)	0.02	0.01	0.02	0.02	0.02	0.02	0.03	0.02	0.02	0.02	0.02	0.02	0.03	0.02	0.01	0.02
linear(y)	0.02	0.01	0.02	0.02	0.02	0.02	0.03	0.03	0.02	0.02	0.02	0.02	0.03	0.03	0.02	0.02
circle	0.05	0.05	0.05	0.05	0.04	0.05	0.06	0.04	0.06	0.06	0.05	0.03	0.06	0.06	0.03	0.02
normal helix	0.12	0.12	0.15	0.10	0.11	0.12	0.14	0.07	0.17	0.16	0.17	0.05	0.16	0.16	0.16	0.02
place-scaled circle	0.31	0.32	0.23	0.22	0.24	0.26	0.18	0.13	0.14	0.14	0.06	0.03	0.14	0.14	0.04	0.02
place-scaled helix (ours)	**0.58**	**0.59**	**0.51**	**0.43**	**0.53**	**0.56**	**0.40**	**0.29**	**0.43**	**0.42**	**0.23**	**0.09**	**0.44**	**0.42**	**0.18**	**0.04**

Table 5: **IIA across tasks and tokenizers.** We measure the IIA averaged over all the places, and show that the Place-Scaled Helix fit shows higher causal relevance.

7 Future Work

Our study focuses on the representation of natural numbers distributed across multiple tokens, but extending this analysis to decimals, integers, and fractions could provide a comprehensive understanding of numerical representations. Our token-wise analysis provides a framework for investigating these classes of numerical values. Furthermore, leveraging the numerical structure to design training strategies to improve numerical reasoning is a promising direction for future work.

8 Conclusion

We study how autoregressive models represent numbers distributed across multiple tokens. Causal steering reveals that, although the final token correlates with the entire number, it does not causally control the representations at other numerical positions, motivating us to analyze each site independently. Within a 20-dimensional number-aligned subspace, we find that each site follows a helical structure whose periods scale with place as $10^{\alpha p}$. This place-scaled helix consistently provides the best fit and highest intervention accuracy compared with linear, logarithmic, circular, and unscaled-helix baselines across single- and multi-digit tokenizers, LLMs and VLMs, and arithmetic tasks. These results suggest a generalized geometric structure underlying numerical representations in autoregressive models, providing a more interpretable view of how models encode numerical values and offering a basis for further understanding and improving numerical reasoning.

Reproducibility Statement

Sections 3–6 describe the model and tokenizer choices, arithmetic task construction, place-scaled helix basis, DAS subspace learning and intervention procedure, cross-validation protocol, and evaluation metrics. Appendix A reports the corresponding analyses for the additional models. The supplementary repository accompanying this submission provides the prompt datasets and generation metadata, model and checkpoint registry, random seeds and experiment configurations, and scripts for activation extraction, causal steering, DAS training, functional-form fitting, IIA evaluation, and figure generation, together with figure-level data and provenance records. Public pretrained checkpoints are referenced by identifier. Large model weights and intermediate activation caches are not redistributed, but the repository documents the required environments and provides cache-or-recompute commands for regenerating them. We will submit the analysis scripts upon acceptance.

References

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt, Dirk Groeneveld, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models, 2024. URL https://arxiv.org/abs/2409.17146.

Jinhong Deng, Limeng Qiao, and Guanglu Wan. Holocount: A holistic visual counting benchmark for mllms, 2026. URL https://arxiv.org/abs/2607.06420.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman. Finding alignments between interpretable causal variables and distributed neural representations, 2024. URL https://arxiv.org/abs/2303.02536.

Rishi Gupta, Mukilan Karuppasamy, Shyam Marjit, Aditay Tripathi, and Anirban Chakraborty. O3slm: Open weight, open data, and open vocabulary sketch-language model, 2025. URL https://arxiv.org/abs/2511.14368.

Hosein Hasani, Amirmohammad Izadi, Fatemeh Askari, Mobin Bagherian, Sadegh Mohammadian, Mohammad Izadi, and Mahdieh Soleymani Baghshah. Understanding counting mechanisms in large language and vision-language models, 2026. URL https://arxiv.org/abs/2511.17699.

Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching, 2024. URL https://arxiv.org/abs/2404.15255.

Marek Kadlčík, Michal Štefánik, Timothee Mickus, Michal Spiegel, and Josef Kuchař. Pretrained language models learn remarkably accurate representations of numbers. *arXiv preprint arXiv:2506.08966*, 2025.

Ivana Kajic and Aida Nematzadeh. Probing representations of numbers in vision and language models. In *SVRHM 2022 Workshop @ NeurIPS*, 2022. URL https://openreview.net/forum?id=01hQQ16Lc9M.

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

Kartik Kuckreja, Muhammad Sohail Danish, Muzammal Naseer, Abhijit Das, Salman Khan, and Fahad Shahbaz Khan. Geochat: Grounded large vision-language model for remote sensing, 2023. URL https://arxiv.org/abs/2311.15826.

Amit Arnold Levy and Mor Geva. Language models encode numbers using digit representations in base 10. *arXiv preprint arXiv:2410.11781*, 2024.

Amit Arnold Levy and Mor Geva. Language models encode numbers using digit representations in base 10, 2025. URL https://arxiv.org/abs/2410.11781.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. URL https://arxiv.org/abs/2304.08485.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2020. URL https://arxiv.org/abs/1802.03426.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2301.05217.

Xingzhou Pang, Yifan Hou, Junling Wang, and Mrinmaya Sachan. Unveiling the visual counting bottleneck in vision-language models, 2026. URL https://arxiv.org/abs/2605.30170.

Michal Štefánik, Timothee Mickus, Marek Kadlčík, Bertram Højer, Michal Spiegel, Raúl Vázquez, Aman Sinha, Josef Kuchař, Philipp Mondorf, and Pontus Stenetorp. Language models learn universal representations of numbers and here's why you should care. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30663–30681, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1415. URL https://aclanthology.org/2026.acl-long.1415/.

Haotong Yang, Yi Hu, Shijia Kang, Zhouchen Lin, and Muhan Zhang. Number cookbook: Number understanding of language models and how to improve it, 2025. URL https://arxiv.org/abs/2411.03766.

Fengting Yuchi, Li Du, and Jason Eisner. Llms know more about numbers than they can say, 2026. URL https://arxiv.org/abs/2602.07812.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building math generalist models through hybrid instruction tuning, 2023. URL https://arxiv.org/abs/2309.05653.

Tianyi Zhou, Deqing Fu, Vatsal Sharan, and Robin Jia. Pre-trained large language models use fourier features to compute addition. *arXiv preprint arXiv:2406.03445*, 2024.

Fangwei Zhu, Damai Dai, and Zhifang Sui. Language models encode the value of numbers linearly, 2024. URL https://arxiv.org/abs/2401.03735.

A Appendix

A.1 Evaluation Setup

We evaluate four 7–8B models: Molmo2-8B and Qwen2.5-7B-Instruct, whose tokenizers represent individual digits, and Llama3.1-8B and Idefics3-8B-Llama3, whose tokenizers group up to three digits per token. All models receive the same frozen arithmetic examples described in Appendix A.4. We use each model's native tokenizer and chat template.

We analyze the number emitted by the model, y_{pred}, and do not condition the primary analyses on exact-answer correctness. Correctness is recorded separately. Activations are extracted at the position that predicts each answer token, with preceding tokens from the model's generated answer supplied when analyzing later answer positions.

For each answer site and candidate layer, we train a 20-dimensional orthonormal DAS subspace for 500 steps using Adam, a batch size of 32, and a learning rate of 10^{-3}. The split is made over distinct emitted values: 70% of the values are used for training and 30% are held out entirely. IIA is evaluated on 384 randomly sampled base–source pairs, together with full-patch, no-intervention, and random-subspace controls. Structural fits are evaluated using 5-fold held-out R^2 over distinct emitted values. Model-specific layers are selected using the causal IIA sweep rather than the structural-fit score.

A.2 Results on Additional Models

The main paper presents the detailed analysis for Molmo2-8B, which uses a single-digit tokenizer, and Llama3.1-8B, which groups up to three digits per token. Here, we repeat the corresponding analyses for the remaining two models in our evaluation: Qwen2.5-7B-Instruct and Idefics3-8B-Llama3. Qwen2.5 provides an additional single-digit-tokenizer model, while Idefics3 provides an additional grouped-tokenizer vision-language model. Unless stated otherwise, these experiments use the same frozen 9-digit addition examples, evaluation procedure, structural baselines, and causal metrics as the corresponding experiments in the main paper.

The following subsections report three complementary results: the locality of causal steering across numerical places, the emergence of place and value organization across layers, and the relationship between structural fit and causal intervention accuracy.

A.2.1 Characteristics of Number Representation

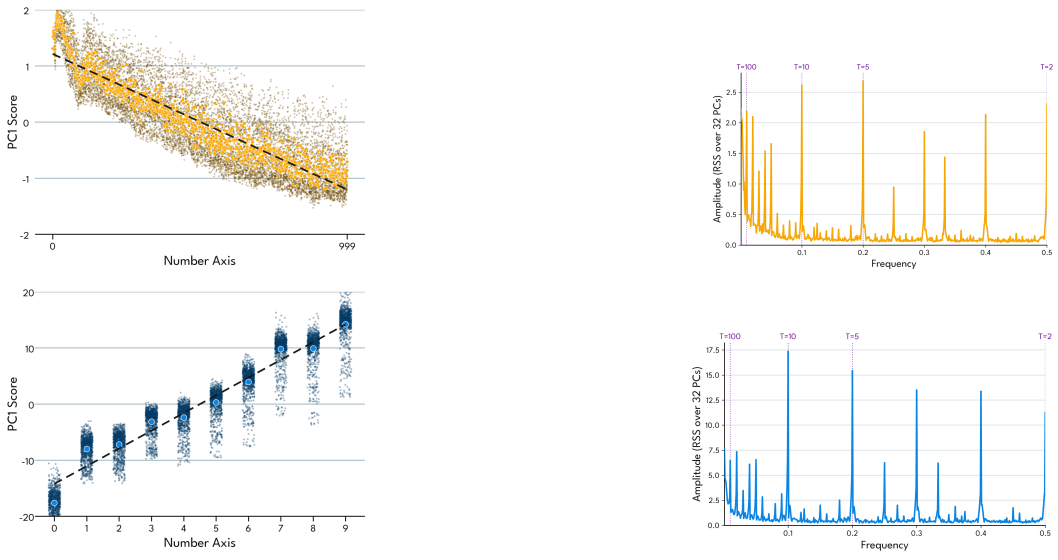

Figure 8: **Characteristics of Number Representation.** Same analysis as Fig. 2 (*left, mid*) for *(Top)* Idefics3-8B and *(Bottom)* Qwen2.5-7B-Instruct. The left column shows the relationship between PC1 and numerical value, while the right column shows the corresponding Fourier spectrum.

A.2.2 STEERABILITY ACROSS PLACES

We repeat the steering experiment of *Fig. 2* (*right*) on the two additional models, shown in *Fig. 9*.

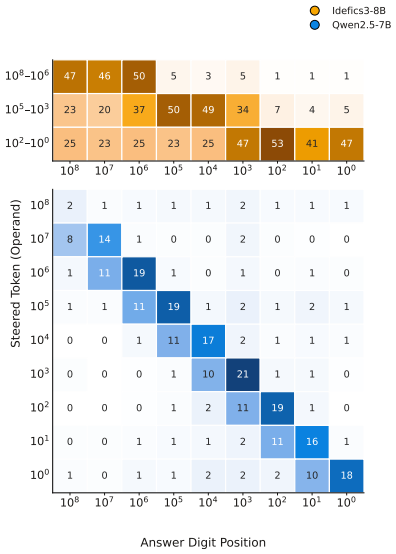

Figure 9: **Steerability on additional models.** Same experiment as *Fig. 2* (*right*). We steer one token of operand a at layer 3 along its digit axis and record which places of the answer change. (*Top*) Idefics3-8B, three digits per token. (*Bottom*) Qwen2.5-7B-Instruct, one digit per token. In both models the steered token's own place and the place to its left change, because of carry over. Idefics3 restates the full equation before answering, so we read its answer after the "=" sign and drop pushes where the operand is no longer read at 9 digits, as for Molmo2. On Idefics3 pushing the units token also moves the higher places in about a quarter of the pushes, which we do not see on Llama3.1. On Qwen2.5 the operand is steerable at every layer up to 20 and not at all from layer 23, consistent with *Fig. 3*.

A.2.3 CLUSTERING OF PLACE AND VALUE

We repeat the analysis of *Fig. 4* on the answer tokens of Qwen2.5 in *Fig. 10* and of Idefics3 in *Fig. 11*. Since the main paper shows this analysis only on Molmo2, we also show it for Llama3.1 in *Fig. 12*, to compare the base LM against Idefics3 which is built on it.

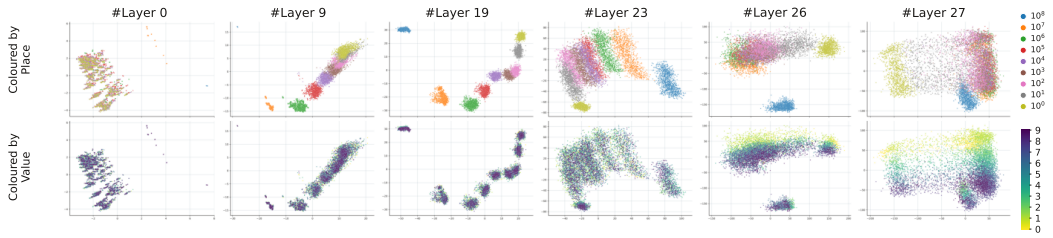

Figure 10: **Clustering of Place and Value on Qwen2.5-7B-Instruct.** Same analysis as *Fig. 4*, on the answer tokens. The place clusters form by layer 9 and stay separated up to layer 22, while the value within each cluster is at chance (η^2 for place rises from 0.39 to 0.91, value stays at 0.01). The value structure appears abruptly at layer 23 and overtakes place at layer 26, showing the same late formation of the number structure as Molmo2.

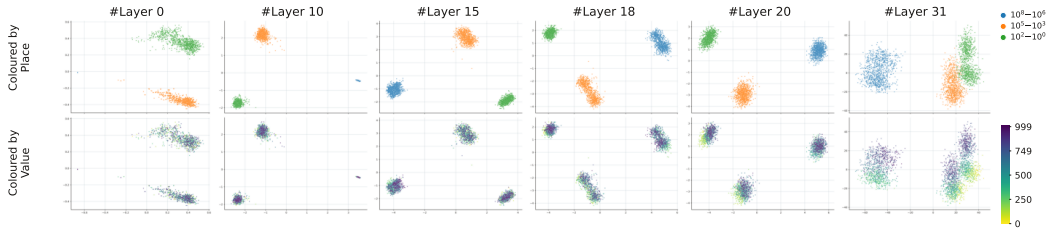

Figure 11: **Clustering of Place and Value on Idefics3-8B.** Same analysis on a multi-digit tokenizer, where a 9-digit answer is three tokens. Each token forms its own place cluster, and the value inside a cluster is the 3-digit group (0–999) that the token holds. The three place clusters are separated from the first layers and stay separated throughout (η^2 for place is 0.70 at layer 0 and above 0.83 from layer 2 to 14). The value within each cluster stays flat up to layer 17 and then forms sharply, with the within-cluster η^2 for value going from 0.64 at layer 17 to 0.78, 0.90 and 0.95 at layers 18, 19 and 20, where it also overtakes place. This is the same layer at which steering the operand stops changing the answer on this model, so the value structure is formed exactly where the model hands over from reading the operands to writing the answer.

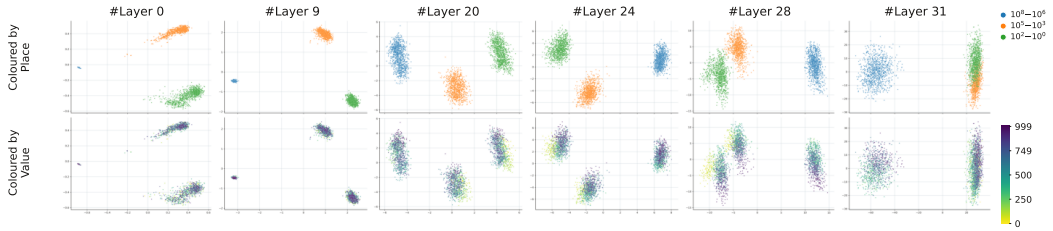

Figure 12: **Clustering of Place and Value on Llama3.1-8B.** Same analysis as *Fig. 11*, on the base LM of Idefics3 without the vision encoder. The three place clusters are separated from the first layer and stay separated throughout (η^2 for place is 0.70 at layer 0 and above 0.84 from layer 2 to 14). The value within each cluster stays flat up to layer 17 and then forms sharply, with the within-cluster η^2 for value going from 0.66 at layer 17 to 0.79, 0.92 and 0.96 at layers 18, 19 and 20, and value overtakes place at layer 19. The value structure is therefore formed at the same layers in Llama3.1 and Idefics3, which again suggests that the vision modality doesn't change the number representation. Unlike Idefics3, the place clusters separate further again in the last layers, with η^2 for place rising from 0.42 at layer 24 to 0.78 at layer 31 as the leading token moves away from the other two.

A.2.4 Emergence of Causality

We repeat the layer-wise analysis from Fig. 7 for the two additional models. For each layer, we report the held-out R^2 of the candidate structures together with the trained-DAS IIA and full-patch ceiling. As shown in Fig. 13, the Place-Scaled Helix provides the strongest fit and becomes causally relevant in the later layers. This transition occurs around layers 20–22 for Idefics3-8B and reaches its maximum at layer 26 for Qwen2.5-7B-Instruct, consistent with the emergence pattern observed for the models in the main paper.

A.3 Noisy subspace Alignment.

PCA and a full-d fit are shaped by everything that varies in the residual stream, not just the number. A DAS subspace is trained causally on the digit, so it shouldn't be. To test this, we pair each clean prompt with a noisy copy. The copy keeps the same operands and the same question, but two unrelated distractors come first: one random image from Holocount dataset Deng et al. (2026) and six sentences of prose with no numbers in them. Any drop in R^2 from clean to noisy therefore measures how much a method depends on variation that has nothing to do with the task.

- **Clean**: "Answer with only the number, nothing else. What is 162+385 = "

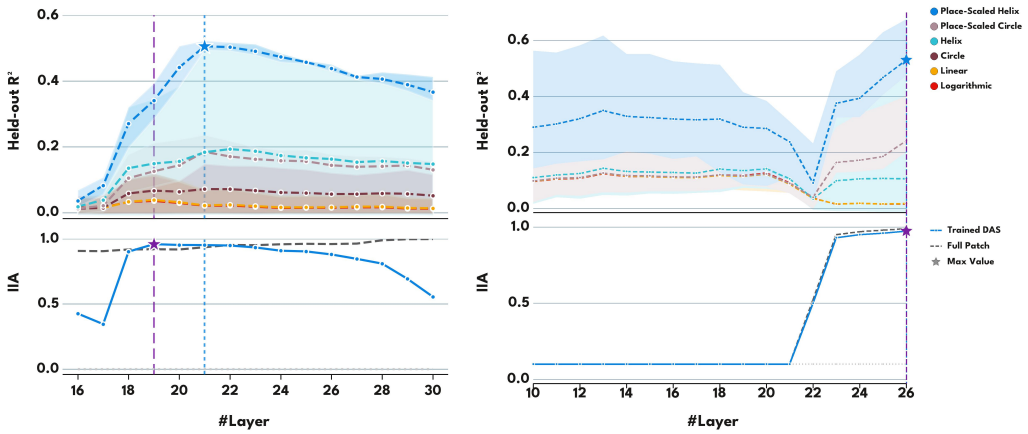

Figure 13: **Emergence of Causality.** This is the same plot as *Fig. 7*. Place-Scaled Helix shows better fit compared to other structures and reaches maximum for (*Left*) Idefics3-8B at layers 20–22 and (*Right*) Qwen2.5-7B-Instruct at layer 26.

- **Noisy**: [image] "Candles flickered on the table as the guests arrived. Rain drifted across the valley while the shepherd gathered his flock. ... The weaver pulled the thread tight and admired the pattern. Answer with only the number, nothing else. What is 162+385 = "

A.4 Dataset Construction and Preprocessing

All arithmetic datasets are generated synthetically using seed 13 and frozen so that every model receives the same examples. Each CSV row contains a unique identifier, operands a and b, target answer, answer width, and complete prompt. The accompanying metadata records the generation parameters, sample statistics, and a SHA-256 checksum. Text experiments use the template `Answer with only the number, nothing else. What is {a}∘{b} =` , where ∘ denotes the arithmetic operation.

Addition. The 3-digit addition set contains 7,955 prompts covering 800 sums from 200 to 999. Both operands are 3-digit numbers, and up to ten distinct ordered operand pairs are sampled without replacement for each sum. The primary multi-token dataset contains 1,000 distinct 9-digit sums sampled without replacement. Both operands are constrained to be 9-digit numbers: for each answer y, operand a is sampled from the feasible range and $b = y - a$. Each answer occurs exactly once.

Other arithmetic operations. The cross-operation experiments use 1,000 9-digit subtraction examples and 1,000 multiplication and division examples with 6-digit answers. For subtraction, both operands are 9-digit numbers satisfying $a - b = y$. For multiplication and division, target answers are constructed so that the digit distribution is approximately uniform at every decimal place. For multiplication, a divisor $b \in [2, 99]$ of y is selected and $a = y/b$. For division, $b \in [2, 99]$ is selected and $a = yb$, ensuring that every division is exact. One- and two-digit values of b are balanced whenever feasible.

Noise and image variants. The noisy addition set is paired one-to-one with the 9-digit addition set and preserves the sample identifier, operands, and answer. Each prompt is augmented with one randomly selected HoloCount images Deng et al. (2026) and six sentences sampled without replacement from a bank of 40 number-free distractor sentences. Thus, the clean and noisy conditions differ only in task-irrelevant visual and textual context.

For the modality experiments, numbers are rendered as centered black text on a 768×768 white canvas using DejaVu Sans Mono. The 9-digit readout condition displays operand a alone and asks the model to return the number in the image; the original addition result is retained separately in the metadata. An analogous 3-digit condition is used for Idefics3, which does not reliably read 9-

digit numbers from images. Images are square-padded and resized before being passed through the model-specific vision processor.

Rows are retained according to the answer width required by each experiment. Model correctness is recorded separately and is not used as a general selection criterion; the representation analyses use the number actually emitted by the model.