

Number Representations in LLMs: A Computational Parallel to Human Perception

H. V. AlquBoj* Hilal AlQuabeh*¹ Velibor Bojkovic*¹

Tatsuya Hiraoka¹ Ahmed Oumar El-Shangiti¹ Munachiso Nwadike¹

Kentaro Inui^{1,2,3}

¹ Mohamed bin Zayed University of Artificial Intelligence (MBZUAI)

² Tohoku University, ³ RIKEN

*Amalgamation of first authors' names.

Abstract

Humans are believed to perceive numbers on a logarithmic mental number line, where smaller values are represented with greater resolution than larger ones. This cognitive bias, supported by neuroscience and behavioral studies, suggests that numerical magnitudes are processed in a sublinear fashion rather than on a uniform linear scale. Inspired by this hypothesis, we investigate whether large language models (LLMs) exhibit a similar logarithmic-like structure in their internal numerical representations. By analyzing how numerical values are encoded across different layers of LLMs, we apply dimensionality reduction techniques such as PCA and PLS followed by geometric regression to uncover latent structures in the learned embeddings. Our findings reveal that the model's numerical representations exhibit sublinear spacing, with distances between values aligning with a logarithmic scale. This suggests that LLMs, much like humans, may encode numbers in a compressed, non-uniform manner¹².

1 Introduction

Large language models (LLMs) have demonstrated impressive capabilities in natural language processing tasks (Touvron et al., 2023; Achiam et al., 2023), yet their internal representations of abstract concepts, i.e., numbers, space, and time, remain largely opaque. Recent research suggests that LLMs construct structured "world models," encoding relationships in ways that can be systematically analyzed (Petroni et al., 2019; Radford et al., 2019). For instance, studies have shown that spatial and geographical information is embedded in low-dimensional subspaces, where model performance correlates with data exposure (Gurnee and

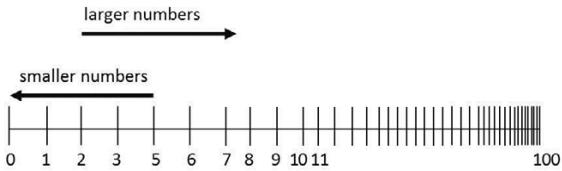


Figure 1: *Logarithmic mental number line hypothesis* asserts that humans innately perceive numbers on a logarithmic scale. Image source (Fritz et al., 2013).

Tegmark, 2024; Godey et al., 2024). Similarly, numerical representation is influenced by tokenization strategies, with base-10 encoding proving more efficient for numeric reasoning tasks than higher-base tokenizations (Zhou et al., 2024).

The *linear hypothesis* of internal representations (Park et al., 2023) posits that concepts in LLMs are structured within geometric, linear subspaces, facilitating interpretability and manipulation. This framework suggests that numerical properties follow systematic, monotonic trends (Heinzerling and Inui, 2024). As a result, it has been commonly assumed that numerical values are represented in a uniform linear fashion (Zhu et al., 2025). However, recent probing studies (Zhu et al., 2025; Levy and Geva, 2024) challenge this assumption, revealing a non-uniform encoding of numbers in LLMs, where precision decreases for larger values. These findings raise questions about how artificial systems internalize numerical representations, particularly in relation to the scaling of numbers. Do LLMs preserve a uniform spacing of numerical values, and if not, what is the nature of their positioning?

Such questions naturally lead to an investigation of whether LLMs encode numerical values in a way that mirrors human cognition, as suggested by the *logarithmic mental number line hypothesis*. This hypothesis posits that humans perceive numerical magnitudes nonlinearly, following a logarithmic rather than a uniform linear scale (see Figure 1). Rooted in psychophysical studies like the Fechner-

¹Code is available at: https://github.com/halquabeh/llm_natural_log

²Correspondence: {hilal.alquabeh, velibor.bojkovic, kentaro.inui}@mbzua.ac.ae

Weber law, this idea is supported by behavioral experiments showing that young children and individuals with limited formal education tend to map numbers logarithmically when placing them on a spatial axis (Fechner, 1860; Dehaene, 2003; Siegler and Opfer, 2003). While formal training shifts numerical perception toward a more linear scale, logarithmic encoding persists in tasks involving estimation and large-number processing (Dehaene et al., 2008; Moeller et al., 2009).

Inspired by this, we investigate whether LLMs encode numerical values in a manner analogous to the human logarithmic mental number line. By analyzing hidden representations across model layers, we examine the geometric structure of numerical magnitudes and their underlying trends. Our approach first employs dimensionality reduction techniques, including Principal Component Analysis (PCA) and Partial Least Squares (PLS), to transform the hidden representations onto a one-dimensional number line, that best fits its dominant numerical features. Second, using Spearman rank coefficient and geometric non-linear regression, we specifically test whether two key properties reminiscent of human numerical cognition (order preservation in representations and a compression effect where distances between consecutive numbers decrease as values increase) emerge in LLMs.

While both PCA and PLS reveal that numerical representations largely reside in a linear subspace, only PCA captures systematic sublinearity, suggesting that simple linear probes³ may overlook the underlying non-uniformity in LLMs’ numerical encoding.

Contributions We summarize our main finding in the following:

- We introduce a methodology for analyzing the geometric structure of number representations, offering a systematic approach to studying numerical abstractions in artificial neural networks.
- We provide empirical evidence that LLMs encode numerical values in a structured yet non-uniform way, revealing systematic compression reminiscent of the human logarithmic mental number line. Our findings refine the linear hypothesis by showing that numerical

³PLS is a linear probe that projects input data onto a lower-dimensional subspace, maximizing covariance with the target.

magnitudes in LLMs are not evenly spaced but follow a structured compression pattern.

2 Related Works

Linearity of internal representations (Park et al., 2023) has been a central assumption in existing research, suggesting that language models encode numerical values in a linear manner. However, Zhu et al. (2025) present a more nuanced perspective. Their analysis of partial number encoding (Appendix F) shows that probing accuracy declines as sequence length increases, with greater difficulty in capturing precise values at larger scales, a pattern reminiscent of logarithmic encoding, where resolution is higher for smaller numbers. Some of the conclusions in Zhu et al. (2025) are that LLMs encode numerical values in their hidden representations, yet linear probes fail to precisely reconstruct these values, as discussed in Zhu et al. (2025, Section 3.1). The authors there suggest that “This phenomenon may indicate that language models use stronger non-linear encoding systems”. Our findings support this claim and further uncover the underlying nature of this non-linearity.

Recent studies such as Levy and Geva (2024); Zhou et al. (2024) show that LLMs rely on base-10 digit-wise representations rather than encoding numbers in a continuous linear space, as revealed through circular probing techniques. While individual digits are accurately reconstructed, performance declines for larger numbers, suggesting a structured rather than holistic encoding. Furthermore, Zhou et al. (2024) demonstrate that LLMs trained on higher-base numeral systems struggle with numerical extrapolation, implying an implicit compressed representation where smaller values have finer granularity—consistent with logarithmic scaling. Collectively, these results align with and further substantiate the hypothesis that LLMs internally represent numbers in a non-uniform, sublinear manner.

Logarithmic functions can appear linear over small local intervals, which may explain why LLMs are often assumed to represent numbers linearly. Consequently, methods like PLS regression and activation patching (Heinzerling and Inui, 2024; El-Shangiti et al., 2024), which analyze small activation variations, may capture local monotonicity while missing the global nonlinear structure. This suggests that reported linear effects could stem from analyzing narrow numerical ranges, whereas

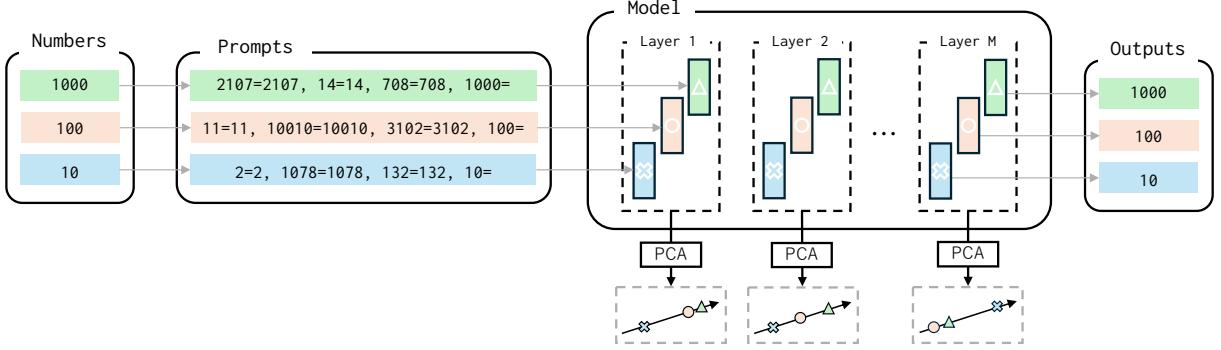


Figure 2: The overall graphical representation of our method. Numbers are passed to the model in form of a prompt and the internal representations are captured from the embeddings corresponding to token '='. At every layer, we perform PCA projections onto one and two dimensional subspaces and pick a layer with highest explained variance (σ^2) score to further analyze monotonicity and scaling of number representations.

a broader examination may reveal an underlying logarithmic representation.

Finally, we also emphasize the difference between our work and prior studies on numerical reasoning (Park et al., 2022; Zhang et al., 2020) which evaluate models’ ability to process explicit numbers rather than probing their internal representations. While Park et al. (2022) focus on tasks like unit conversion and range detection, Zhang et al. (2020) examine numerical magnitude in common sense reasoning. Unlike these works, our study investigates the spatial structure of numerical representations within hidden states and how this encoding generalizes across scales.

3 Methodology

3.1 General setup

The *logarithmic mental number line hypothesis* (Dehaene et al., 2008) suggests that humans innately perceive numerical magnitudes on a logarithmic scale rather than a linear one. Formally, if we denote the internal mapping of numbers to their cognitive representation as f_H , the hypothesis asserts that f_H is approximately logarithmic. While the exact nature of f_H remains elusive, we adopt the guiding principle $f_H \equiv \log$ in our experiments.

On the other side, Large language models, like LLaMA-2, process inputs by mapping them into a high-dimensional representation space, where each input x (e.g., a number) is transformed into an internal representation $f(x) \in \mathbb{R}^d$. Analyzing the geometry of these representations across a set of inputs \mathcal{X} can reveal how the model organizes and reasons about them, shedding light on emergent properties such as whether the model encodes

numerical values along a number line or exhibits cognitive-like patterns, such as sublinear scaling.

Our goal is to study the properties of the function f_{LLM} , which serves as a counterpart of the human cognitive mapping f_H described above. Specifically, we examine whether f_{LLM} preserves the natural ordering of numbers and how it transforms their magnitudes.

3.2 Definition of f_{LLM}

To analyze the structure of hidden representations, we apply a projection $T : \mathbb{R}^d \rightarrow \mathbb{R}^p$, $p = 1, 2$, obtained from techniques such as Principal Component Analysis (PCA) or Partial Least Squares (PLS). Then, our function is given by

$$f_{LLM}(x) := T(f(x)), \quad (1)$$

where f is a map between the input number and the corresponding internal representation in the model (see Sections 4 and 5 how f is defined in various settings).

For any two inputs $x, y \in \mathcal{X}$, we define the distance between their projections following the mapping T using Euclidean norm as

$$d(x, y) = \|f_{LLM}(x) - f_{LLM}(y)\|. \quad (2)$$

3.3 Abstract Number Line and Monotonicity Metric

If the projection dimension is $p = 1$, one-dimensional embedding of numerical inputs forms a *number line* if the projections preserve monotonicity (resp. reverse monotonicity), i.e., for $x_1 < x_2$ (resp. $x_1 > x_2$), we have $f_{LLM}(x_1) < f_{LLM}(x_2)$. This ensures that the natural order of numerical values is maintained in the representation space.

To measure monotonicity properties of the function f_{LLM} we use Spearman rank correlation that we briefly describe next. Let $X, Y \in \mathbb{R}^n$ be two real n -dimensional vectors and let $R(X)$ (resp. $R(Y)$) denote an n -dimensional vector obtained from X (resp. Y) where the entries are substituted with their ranks in the sequence of sorted entries of X (resp. Y). Then, Spearman rank correlation coefficients (usually denoted by ρ) is given by:

$$\rho = \frac{\text{Cov}(R(X), R(Y))}{\sigma(R(X)) \cdot \sigma(R(Y))}, \quad (3)$$

where $\text{Cov}(R(X), R(Y))$ is the covariance between rank vectors $R(X)$ and $R(Y)$, while $\sigma(R(X))$ and $\sigma(R(Y))$ are their respective standard deviations.

Spearman coefficient ρ is a nonparametric measure for the alignment of the two vectors. Loosely speaking, the coefficient assesses if the increment in one variable corresponds to the increase (or decrease) of the other. In particular, unlike Pearson coefficient which takes into account the value of the changes, Spearman's ρ takes into account only the sign of the changes.

3.4 Scaling Rate Index

For a monotonically increasing sequence of positive real numbers x_1, \dots, x_n , we introduce the Scaling Rate Index to measure the rate at which numbers grow in magnitude. More precisely, we seek for positive real constants α and β that minimize the following objective function:

$$\sum_{i=1}^{n-1} |(x_{i+1} - x_i) - \alpha \cdot \beta^i|^2. \quad (4)$$

In particular, if $\beta > 1$, the difference between two consecutive members x_i and x_{i+1} is expected to increase with i . In other words, $x_{i+1} - x_i$ is expected to be smaller than $x_{i+2} - x_{i+1}$. Such sequences are commonly known as *convex sequences* (Rockafellar, 2015) and the growth of x_i is exponential (superlinear) in i . An example of such a sequence that is relevant to our study is that defined with $x_i := 10^i$. Then, $x_{i+1} - x_i = 9 \cdot 10^i$ and we can take $\alpha = 9$ and $\beta = 10$.

If $\beta = 1$, expression (4) indicates that the difference between two consecutive members x_i and x_{i+1} is approximately constant, i.e. the sequence is approximately linearly increasing. For example, one may take the sequence $x_i := i$, for which $\alpha = 1 = \beta$.

Finally, $\beta < 1$ implies that the difference between consecutive members is steadily decreasing, $x_{i+1} - x_i$ is expected to be greater than $x_{i+2} - x_{i+1}$. Such sequences are commonly known as *concave sequences* (Rockafellar, 2015), and the growth of the sequence is exponentially decreasing (sublinear). An example of such a sequence is given by $x_i = 1 - \frac{1}{10^i}$, with $\alpha = 9$ and $\beta = \frac{1}{10}$.

4 Experiment 1: Identifying number line using contextualized numbers

Setup. To systematically probe the model's numerical representations, we partition numbers into logarithmically spaced groups:

$$G_1 = \{1, 2, \dots, 20\}, \quad G_{i+1} = \{10^i - 19, \dots, 10^i + 20\}, \quad i \geq 2. \quad (5)$$

These groups ensure that larger group indices correspond to numerical magnitudes that increase exponentially with index, reflecting the logarithmic nature of the mental number line hypothesis.

To analyze the embeddings of numbers, to every number $x \in G_i$ we assign the following prompt:

$$x \leftarrow a=a, b=b, c=c, x= \quad (6)$$

where a , b , and c are randomly generated numbers from the groups G_i . This prompt structure is designed to provide the model with contextual examples, encouraging it to invoke the number x in model's hidden states representations (see Figure 2). Such approaches have been used in prior work to probe contextual representations in language models Srivastava et al. (2024).

The hidden state representation $f(x) \in \mathbb{R}^d$ is extracted from a designated layer of the model from the last token in the prompt, e.g. the '=' token. We use only those x for which the generated output of the model is x itself.

To ensure a representative sampling, we randomly select k numbers from each group G_i . These sampled numbers collectively form a dataset, denoted as \mathcal{X} , which serves as the basis for our analysis. The set of hidden state representations, $\{f(x)\}_{x \in \mathcal{X}}$, is then aggregated and analyzed to investigate patterns and properties in the embedding space.

To control for potential biases introduced by tokenization (where larger numbers often span more tokens) we conduct a complementary experiment

using non-numerical sequences. Instead of numerical inputs, we construct sequences of random letters with lengths corresponding to the tokenized representations of numbers. The letter sequences are grouped to their lengths so that the grouping approximately matches one of the numbers, and the prompts corresponding to specific letter sequences are designed in a similar fashion as for the numbers (6). This setting allows us to compare any observed structural patterns between the number representations and letter representations. By doing so, we can determine whether the model truly encodes numerical magnitude or if it is simply responding to surface-level features of the input.

Motivation. The goal of this experiment is twofold. We first investigate whether LLMs encode numerical values in context along a *monotonic number line* in their internal representation space. Second, we test whether this number line exhibits *sublinear scaling*, similar to human cognitive representations of numbers.

Methodology. For these purposes, we use:

- PCA nad PLS. After the set of hidden state representations is aggregated, we further project it into a one-dimensional space using PCA or PLS methods. In particular, for PLS we take the numbers themselves to form the target vectors, while for letters, we consider the letter sequence as a base 26 representation of a number (with random but fixed assignment of values to the letters), and use this number as the corresponding target.
- Monotonicity metric. We apply it on a sequence x_1, \dots, x_n of all the numbers in the union of groups G_j defined in (5), and their respective projections $f_{\text{LLM}}(x_1), \dots, f_{\text{LLM}}(x_n)$. Spearman rank coefficient will tell us whether the model preserves natural ordering of the numbers.
- Scaling Rate Index. The initial centers of G_i , given by $x_i = 10^i$, form a convex, exponentially growing sequence characterized by parameters $\alpha = 9$ and $\beta = 10$. These numbers serve as representative scales of the numbers within each group.

To obtain a robust estimate of how these scales are preserved in the projections under f_{LLM} , we compute the expectation of projections in each group. Specifically, for SRI analysis, we define the sequence

$$\bar{f}_{\text{LLM}}(i) = \mathbb{E}_{x \in G_i} [f_{\text{LLM}}(x)], \quad (7)$$

and fit positive constants α and β such that $\bar{f}_{\text{LLM}}(i) \approx \alpha \cdot \beta^i$.

In particular, the mapping

$$10^i \mapsto \bar{f}_{\text{LLM}}(i) \quad (8)$$

allows us to examine how does f_{LLM} scales numerical magnitudes. The fundamental question we seek to answer is: **What is the nature of the function f_{LLM} (logarithmic, linear, or exponential)?**

To answer this question, we analyze the scaling factor β in the fitted exponential model⁴. In the following, we explain how different values of β correspond to the underlying properties of f_{LLM} .

- If $\beta > 1$, the sequence $\bar{f}_{\text{LLM}}(i)$ is convex and exponentially increasing. This means f_{LLM} maps an exponentially increasing sequence to another exponentially increasing sequence:

$$9 \cdot 10^i \mapsto \alpha \cdot \beta^i = \alpha \cdot 10^{(\log_{10} \beta) \cdot i}.$$

Thus, f_{LLM} preserves the original spacing of numbers, albeit with a scaling factor $\log_{10} \beta > 0$.

- If $\beta = 1$, the sequence $\bar{f}_{\text{LLM}}(i)$ is linearly increasing, meaning f_{LLM} takes the form:

$$10^i \mapsto \alpha \cdot i = \alpha \cdot \log_{10} 10^i.$$

In this case, f_{LLM} exhibits logarithmic scaling.

- If $\beta < 1$, the sequence $\bar{f}_{\text{LLM}}(i)$ is concave, exponentially decaying. Here, f_{LLM} follows:

$$10^i \mapsto \alpha \cdot \beta^i = \alpha \cdot 10^{-(\log_{10} \frac{1}{\beta}) \cdot i}.$$

Thus f_{LLM} is a *sub-logarithmic* mapping.

Results. The results reveal distinct yet consistent patterns in how different models encode numerical and alphabetical structures, with variations across layers (Tables 1 and 2). Despite these variations, similar trends emerge across the models, leading to consistent conclusions about their processing of numerical values (please refer to appendix A for experimental details).

Numerical vs. Symbolic Representations. First key finding is that numerical embeddings exhibit a significantly higher explained variance

⁴We can disregard α from the analysis since it does not influence the scaling but merely introduces a bias.

Model	Group	Layer	$\rho \pm \text{std}$	$\beta \pm \text{std}$	$\sigma^2 \pm \text{std}$
LLaMA-2-7B	Numbers	3	0.97 ± 0.00	0.83 ± 0.06	0.60 ± 0.01
	Letters	1	0.45 ± 0.00	1.21 ± 0.00	0.24 ± 0.00
Pythia-2.8B	Numbers	8	0.94 ± 0.01	0.54 ± 0.01	0.31 ± 0.01
	Letters	11	0.89 ± 0.01	0.53 ± 0.10	0.16 ± 0.01
GPT-2-L	Numbers	18	0.95 ± 0.00	0.58 ± 0.02	0.32 ± 0.00
	Letters	5	0.11 ± 0.05	0.80 ± 0.42	0.21 ± 0.01
Mistral-7B	Numbers	3	0.96 ± 0.00	1.05 ± 0.00	0.44 ± 0.00
	Letters	14	0.89 ± 0.00	0.60 ± 0.00	0.22 ± 0.00
LLaMA-3.1-8B	Numbers	1	0.41 ± 0.04	1.14 ± 0.05	0.48 ± 0.01
	Letters	1	0.56 ± 0.00	0.16 ± 0.07	0.19 ± 0.01
LLaMA-3.2-Instruct-1B	Numbers	4	0.93 ± 0.02	1.33 ± 0.12	0.35 ± 0.01
	Letters	1	0.57 ± 0.06	0.47 ± 0.08	0.17 ± 0.00

Table 1: Comparison of several models on Numbers and Letters groups, evaluated using three metrics: ρ , β , and Explained Variance (σ^2). Results are reported for the layer with the highest σ^2 score. Standard deviations are included.

Model	Group	Layer	$\rho \pm \text{std}$	$\beta \pm \text{std}$	$R^2 \pm \text{std}$
Llama-3.2-1B-Instruct	Numbers	6	0.91 ± 0.00	1.93 ± 0.05	0.68 ± 0.01
	Letters	10	0.93 ± 0.00	0.97 ± 0.03	0.45 ± 0.03
Pythia-2.8b	Numbers	1	0.78 ± 0.02	4.65 ± 1.32	0.71 ± 0.01
	Letters	20	0.90 ± 0.01	0.95 ± 0.11	0.46 ± 0.04
GPT2-L	Numbers	17	0.96 ± 0.01	1.15 ± 0.09	0.67 ± 0.03
	Letters	33	0.81 ± 0.03	0.93 ± 0.04	0.44 ± 0.01
Llama-2-7b	Numbers	5	0.93 ± 0.00	2.62 ± 0.00	0.81 ± 0.00
	Letters	27	0.88 ± 0.03	0.91 ± 0.02	0.45 ± 0.01
Mistral-7B-v0.1	Numbers	7	0.88 ± 0.00	14.87 ± 8.28	0.81 ± 0.00
	Letters	29	0.86 ± 0.00	1.62 ± 0.00	0.63 ± 0.00
Llama-3.1-8B	Numbers	4	0.93 ± 0.01	2.00 ± 0.01	0.73 ± 0.01
	Letters	16	0.93 ± 0.01	0.88 ± 0.06	0.45 ± 0.02

Table 2: Comparison of several models on Numbers and Letters groups, evaluated using three metrics: ρ , β , and R^2 . Results are reported for the layer with the highest R^2 . Standard deviations are included.

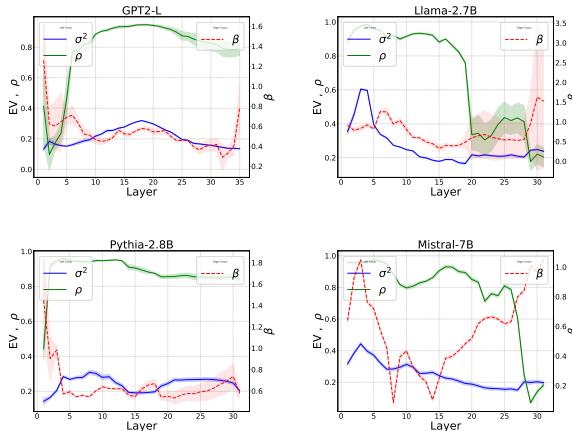


Figure 3: Layer-wise analysis of four models on numerical groups, showing explained variance (σ^2), monotonicity (ρ), and Scaling Rate Index (β). The layer with maximum σ^2 aligns with peak ρ , indicating optimal numerical encoding.

(σ^2 in Table 1 and R^2 in Table 2) in the one-

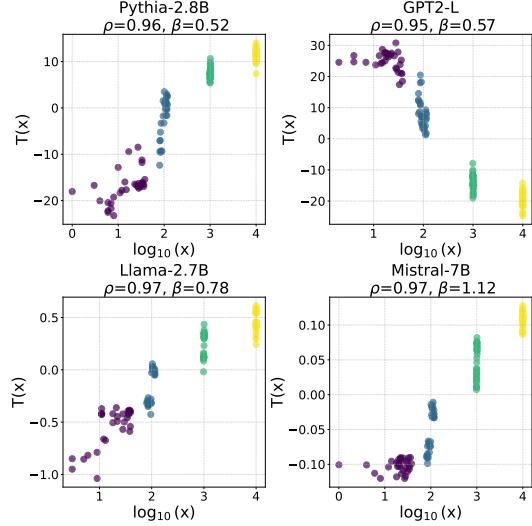


Figure 4: Projections of numerical representations (y-axis) against their log-scaled magnitudes (x-axis) for the layer with the highest explained variance in four models. Sublinearity and monotonicity (ρ) are indicated above each subfigure, demonstrating consistent sublinear trends and strong monotonic relationships across models.

dimensional PCA and PLS transformations compared to letter-based embeddings (refer to Methodology). This suggests that numbers naturally align along a one-dimensional manifold—akin to a number line—while random sequences of letters do not display the same structured behavior.

Furthermore, the monotonicity metric (ρ) consistently shows higher values for numerical data, with most models achieving $\rho > 0.9$ in both PCA and PLS analyses. This supports the idea that numerical representations are not only structured, but also maintain a well-ordered progression across layers.

The resulting projections obtained using PCA for the numerical and letters groups are visualized in Figures 4 and 5, respectively.

Sublinearity in Numerical Representations. The sublinearity coefficient (β) derived from PCA projections reveals notable differences across models. Some, such as LLaMA-2-7B, Pythia, and GPT-2 Large, exhibit strong sublinear (sublogarithmic) scaling with $\beta < 1$, indicating that embedding distances grow at a diminishing rate. In contrast, models like Mistral show a nearly logarithmic trend ($\beta \approx 1$), while others approach a more linear spacing pattern with higher β values.

Layer-Wise Dynamics. Since Table 1 reports values from the layer with the highest explained variance, interpretation requires caution—other lay-

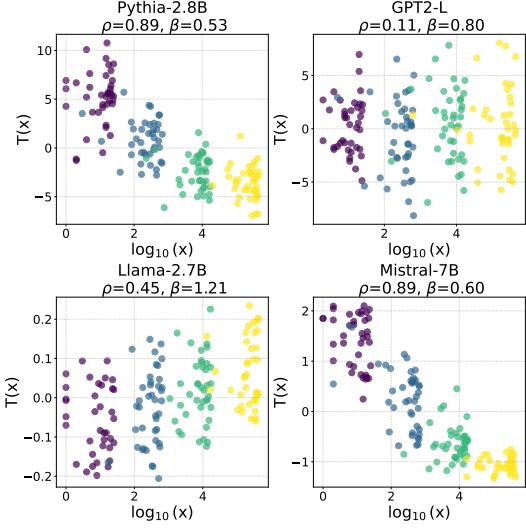


Figure 5: Projections of letters representations (y-axis) against their log-scaled magnitudes (x-axis) assigned proportional to their length, for the layer with the highest explained variance in four models. Sublinearity and monotonicity (ρ) are indicated above each subfigure, demonstrating consistent sublinear trends and strong monotonic relationships across models.

ers with comparable σ^2 values may exhibit similar trends. Figure 3 provides a layer-wise analysis for four models, demonstrating how sublinearity evolves across different depths.

PCA vs PLS. The PLS method achieves high monotonicity (ρ) and explained variance (R^2) but exhibits lower sublinearity compared to PCA. This discrepancy arises because PLS operates as a supervised linear probe, where the regression target (e.g., numerical values) directly influences the projection. This process distorts the intrinsic spacing between points, as PLS prioritizes maximizing covariance with the target over preserving the original geometric structure. In contrast, PCA, being unsupervised, retains the relative spacing of data points in the latent space, better capturing the underlying sublinear trends. This distinction is evident in tables 1 and 2: PCA consistently reveals stronger sublinearity, while PLS achieves higher R^2 and ρ by aligning the projection with the target variable.

Notably, this aligns with findings in Zhu et al. (2025), where a linear probe failed to adequately capture the non-linear scaling of hidden states, particularly for larger numbers, where non-linearity becomes more pronounced. Our work explicitly quantify sublinearity using the Scaling Rate Index (SRI, β), which directly measures the rate of scaling in the latent space. This allows us to better

capture the true geometric organization of numerical representations, especially in regimes where non-linear effects dominate.

Ablation study. Finally, we perform an ablation study to examine the dependence of our results on the number of examples in the prompt for both numerical and alphabetical datasets. Figure 6 shows that the metrics exhibit greater stability for numerical data compared to alphabetical data, indicating that the model processes numerical information more consistently, while alphabetical representations are more sensitive to prompt variations.

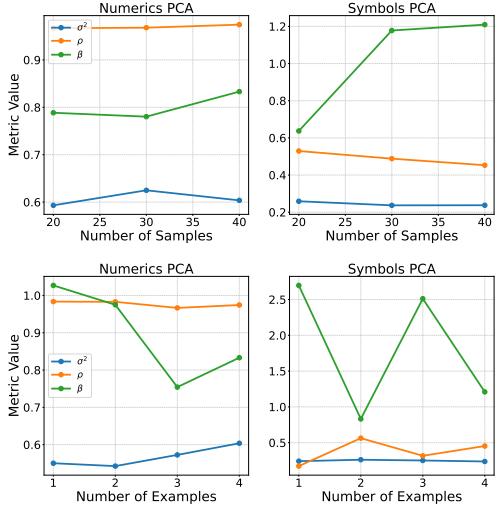


Figure 6: Top row: Change in metrics with respect to the number of samples in the set \mathcal{X} . Bottom row: Change in metrics with respect to the number of in-context examples in the prompt. Left column corresponds to the **numbers** group, and the right column corresponds to the **letters** group. Sublinearity and monotonicity trends are highlighted for each case.

5 Experiment 2: Identifying number line using real-world tasks

Setup. In the previous experiment (Section 4) we created an artificial experimental setting to test our hypothesis. In this experiment, however, we want to further validate our hypothesis using real-world data. We collect names of celebrities along with their birth years and population of different cities/countries from Wikidata (Vrandečić and Krötzsch, 2014). The task here is to investigate for similar patterns and observations seen in the previous experiment.

Motivation. The goal of this experiment is to investigate how LLMs internally represent numerical

values in real-world contexts, specifically focusing on the monotonicity and scaling of these representations. By analyzing the hidden states, we aim to uncover whether the models encode numerical information in a structured and interpretable manner.

Methodology. The experimental setup for this experiment is as follows:

- **Prompting the Model:** We prompt the model to provide the exact birth year or population size for each entity in our dataset, which consists of 1K samples. An example of a prompt would be “What is the population of [country]?”
- **Collecting Model Outputs:** We collect the LLM’s output answers, and filter out non-numerical and incorrect responses.

- **Extracting Hidden States:** We extract the hidden state corresponding to the question mark token at each model layer⁵.

- **Training PLS Models:** We train one- and two-component PLS models on the extracted hidden states to predict the birth years or population sizes of the entities. This is performed for two LLMs: *Llama-3.1-8B* and *Llama-3.2-1B*.

Results. The results, as shown in table 3, demonstrate a clear distinction between the two tasks, but also between the models.

For the *birth year* task, model *Llama-3.1-8B* exhibit strong trends, with high monotonicity (ρ) and (R^2), while having low SRI (β), hence high compression. This indicates that the internal representations of birth years are well-structured and predictive, aligning with our expectations for numerical encoding in LLMs. On the other side, *Llama-3.2-1B*) shows low monotonicity score, hence the β factor is not informative. We attribute the low monotonicity score to the non-structured internal representations of lower birth years, as can be seen in Figure 7.

For the *population size* task, both models display weaker monotonicity and lower R^2 , suggesting population sizes are encoded less systematically. Unlike birth years, population figures are more context-dependent, influenced by geopolitical changes, reporting inconsistencies, and approximate expressions in text. Consequently, the low monotonicity makes the scaling ratio β unreliable.

Finally Figure 7 shows the examples of one and two PLS projections for two models, for birth-year

⁵We divided the hidden states into four equally sized groups, ranging from the minimum to the maximum answers, to facilitate the calculation of the Scaling Rate Index (SRI).

dataset.

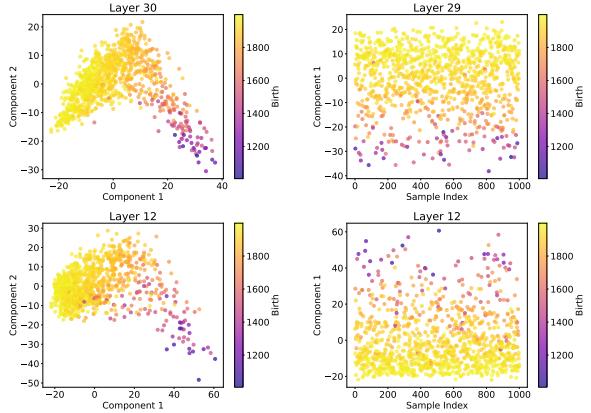


Figure 7: Visualization of PLS models trained on Llama-3.1-8B (top row) and Llama-3.2-1B (bottom row) model activations to predict entities’ birth years using one and two dimensional PLS respectively. Each subfigure represents the layer with the highest R^2 score for one- and two-component PLS models.

Model	Dataset	Layer	ρ	β	R^2
Llama 3.1 8B	Birth	29	0.84	0.50	0.63
	Population	9	0.63	2.48	0.08
Llama 3.2 1B	Birth	12	0.03	0.72	0.61
	Population	10	0.62	2.69	0.10

Table 3: Results for Llama models evaluated on the Birth and Population datasets. Results are reported for the layer with the highest R^2 , highlighting the relationship between scaling rate (β), monotonicity (ρ), and model performance.

6 Conclusion

Inspired by the logarithmic compression in human numerical cognition, we investigate whether LLMs encode numerical values analogously. By analyzing hidden states across layers, we employ dimensionality reduction techniques (PCA and PLS) and geometric regression to test for two key properties: (1) order preservation and (2) sublinear compression, where distances between consecutive numbers decrease as values increase. Our results reveal that while both PCA and PLS identify numerical representations in a linear subspace, only PCA captures systematic sublinearity. This indicates that linear probes like PLS, which optimize for covariance with the target, may obscure the underlying non-uniform structure. Our findings suggest that LLMs encode numerical values with structured compression, akin to the human mental number line, but

this is only detectable through methods like PCA that preserve geometric relationships.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Stanislas Dehaene. 2003. The neural basis of the weber–fechner law: a logarithmic mental number line. *Trends in cognitive sciences*, 7(4):145–147.
- Stanislas Dehaene, Véronique Izard, Elizabeth Spelke, and Pierre Pica. 2008. Log or linear? distinct intuitions of the number scale in western and amazonian indigene cultures. *science*, 320(5880):1217–1220.
- Ahmed Oumar El-Shangiti, Tatsuya Hiraoka, Hilal AlQuabeh, Benjamin Heinzerling, and Kentaro Inui. 2024. [The geometry of numerical reasoning: Language models compare numeric properties in linear subspaces](#). *Preprint*, arXiv:2410.13194.
- Gustav Theodor Fechner. 1860. *Elemente der psychophysik*, volume 2. Breitkopf u. Härtel.
- Annemarie Fritz, Antje Ehlert, and Lars Balzer. 2013. Development of mathematical concepts as basis for an elaborated mathematical understanding. *South African Journal of Childhood Education*, 3(1):38–67.
- Nathan Godey, Éric Villemonte De La Clergerie, and Benoît Sagot. 2024. On the scaling laws of geographical representation in language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 12416–12422.
- Wes Gurnee and Max Tegmark. 2024. Language models represent space and time. In *The Twelfth International Conference on Learning Representations*.
- Benjamin Heinzerling and Kentaro Inui. 2024. Monotonic representation of numeric properties in language models. *arXiv preprint arXiv:2403.10381*.
- Fengqing Jiang. 2024. Identifying and mitigating vulnerabilities in llm-integrated applications. Master’s thesis, University of Washington.
- Amit Arnold Levy and Mor Geva. 2024. Language models encode numbers using digit representations in base 10. *arXiv preprint arXiv:2410.11781*.
- Korbinian Moeller, Silvia Pixner, Liane Kaufmann, and Hans-Christoph Nuerk. 2009. Children’s early mental number line: Logarithmic or decomposed linear? *Journal of experimental child psychology*, 103(4):503–515.
- Kiho Park, Yo Joong Choe, and Victor Veitch. 2023. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*.
- Sungjin Park, Seungwoo Ryu, and Edward Choi. 2022. [Do language models understand measurements?](#) *Preprint*, arXiv:2210.12694.
- Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Ralph Tyrell Rockafellar. 2015. *Convex analysis*. Princeton university press.
- Robert S Siegler and John E Opfer. 2003. The development of numerical estimation: Evidence for multiple representations of numerical quantity. *Psychological science*, 14(3):237–250.
- Pragya Srivastava, Satvik Golechha, Amit Deshpande, and Amit Sharma. 2024. [NICE: To optimize in-context examples or not?](#) In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5494–5510, Bangkok, Thailand. Association for Computational Linguistics.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Denny Vrandečić and Markus Krötzsch. 2014. [Wikidata: A free collaborative knowledgebase](#). *Communications of the ACM*, 57:78–85.
- Xikun Zhang, Deepak Ramachandran, Ian Tenney, Yanai Lazar, and Dan Roth. 2020. Do language embeddings capture scales? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 292–299.
- Zhejian Zhou, Jiayu Wang, Dahua Lin, and Kai Chen. 2024. Scaling behavior for large language models regarding numeral systems: An example using pythia. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3806–3820.
- Fangwei Zhu, Damai Dai, and Zhifang Sui. 2025. Language models encode the value of numbers linearly. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 693–709.

A Experimental details

All experiments were performed using an NVIDIA A6000 GPU for accelerated computation. The models were implemented in Python and imported from Huggingface with PyTorch, and standard libraries like NumPy and Matplotlib were used for data processing and visualization. We evaluated the following models:

Model	Variants	Ref.
Pythia	2.8B	(Touvron et al., 2023)
LLaMA	2.7B, 3.1-8B, 3.2-1B	(Touvron et al., 2023)
GPT-2	Large-1.5B	(Radford et al., 2019)
Mistral	7B	(Jiang, 2024)

Table 4: Models evaluated in the experiments.

Whenever possible, results were reported as the average of three runs, along with the standard deviation (std). For experiments where repeated runs were not feasible, the random seed was fixed to 42 to ensure reproducibility.

B Additional experiments

B.1 Layer-wise PLS analysis

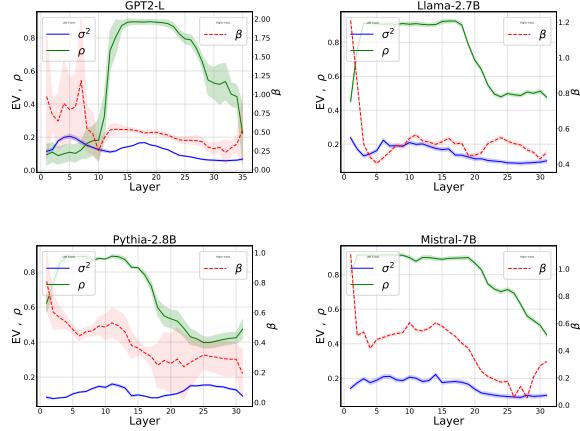


Figure 8: Layer-wise analysis of four models on letters groups, showing explained variance (σ^2), monotonicity (ρ), and Scaling Rate Index (β).

B.2 Birth year and population datasets projections in all layers

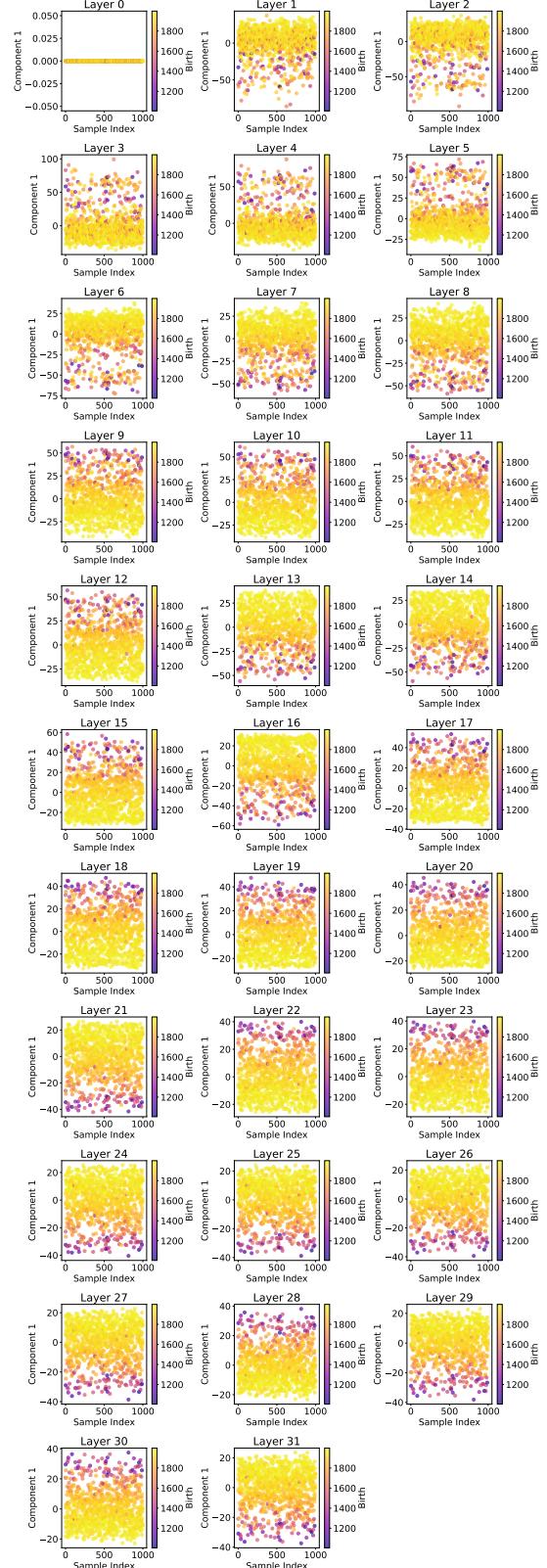


Figure 9: One component PLS model trained on Llama-3.1-8B instruct model activations to predict entities' birth year.

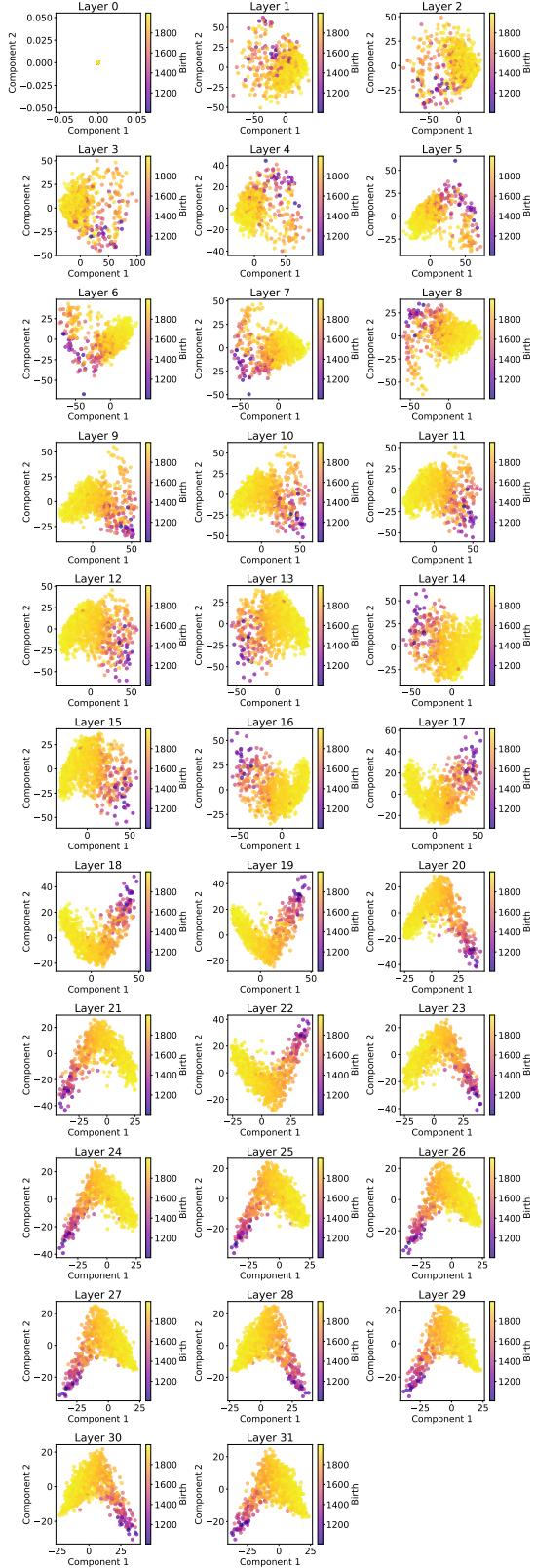


Figure 10: Two components PLS model trained on Llama-3.1-8B instruct model activations to predict entities' birth year.

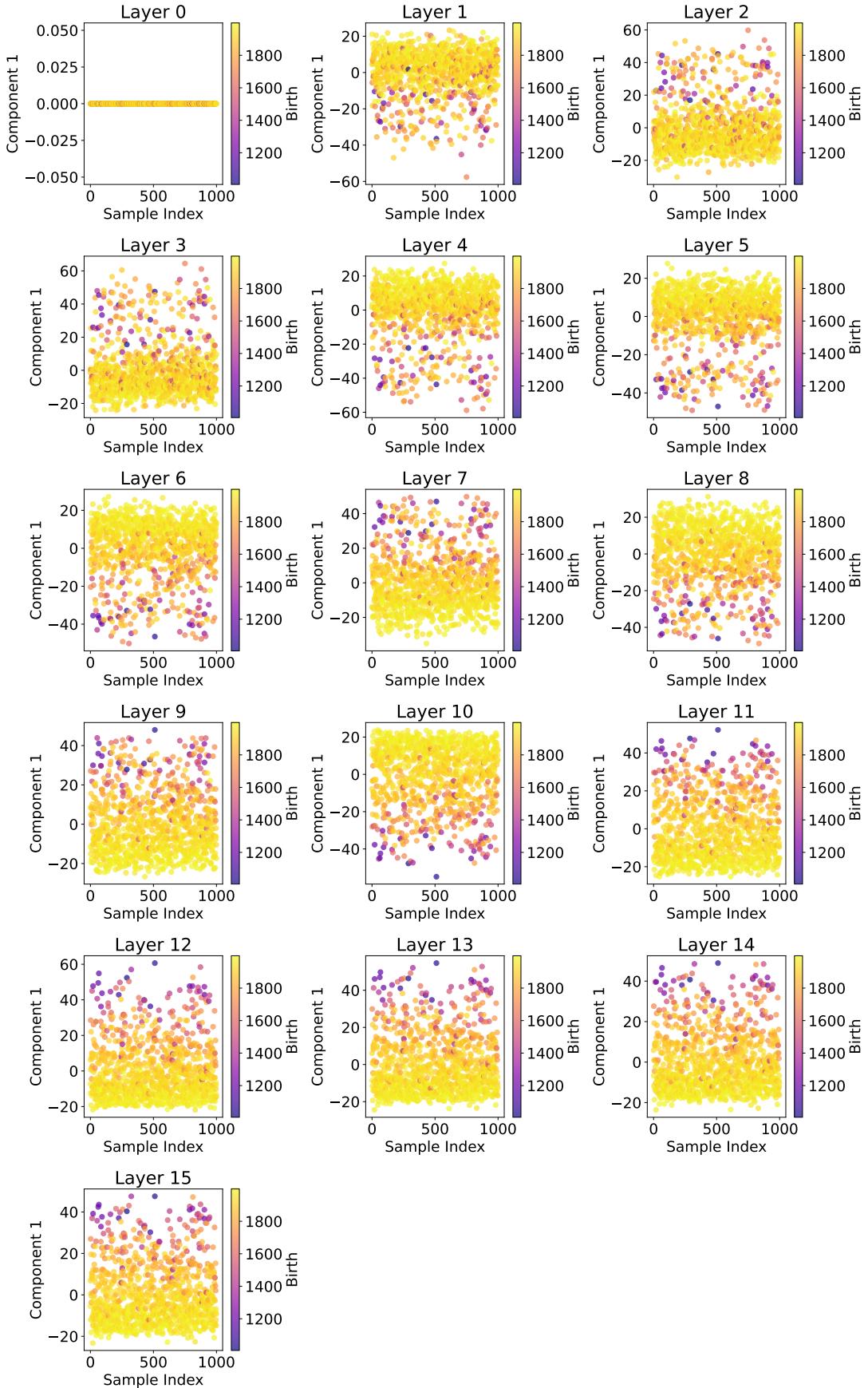


Figure 11: One component PLS model trained on Llama-3.2-1B instruct model activations to predict entities' birth year.

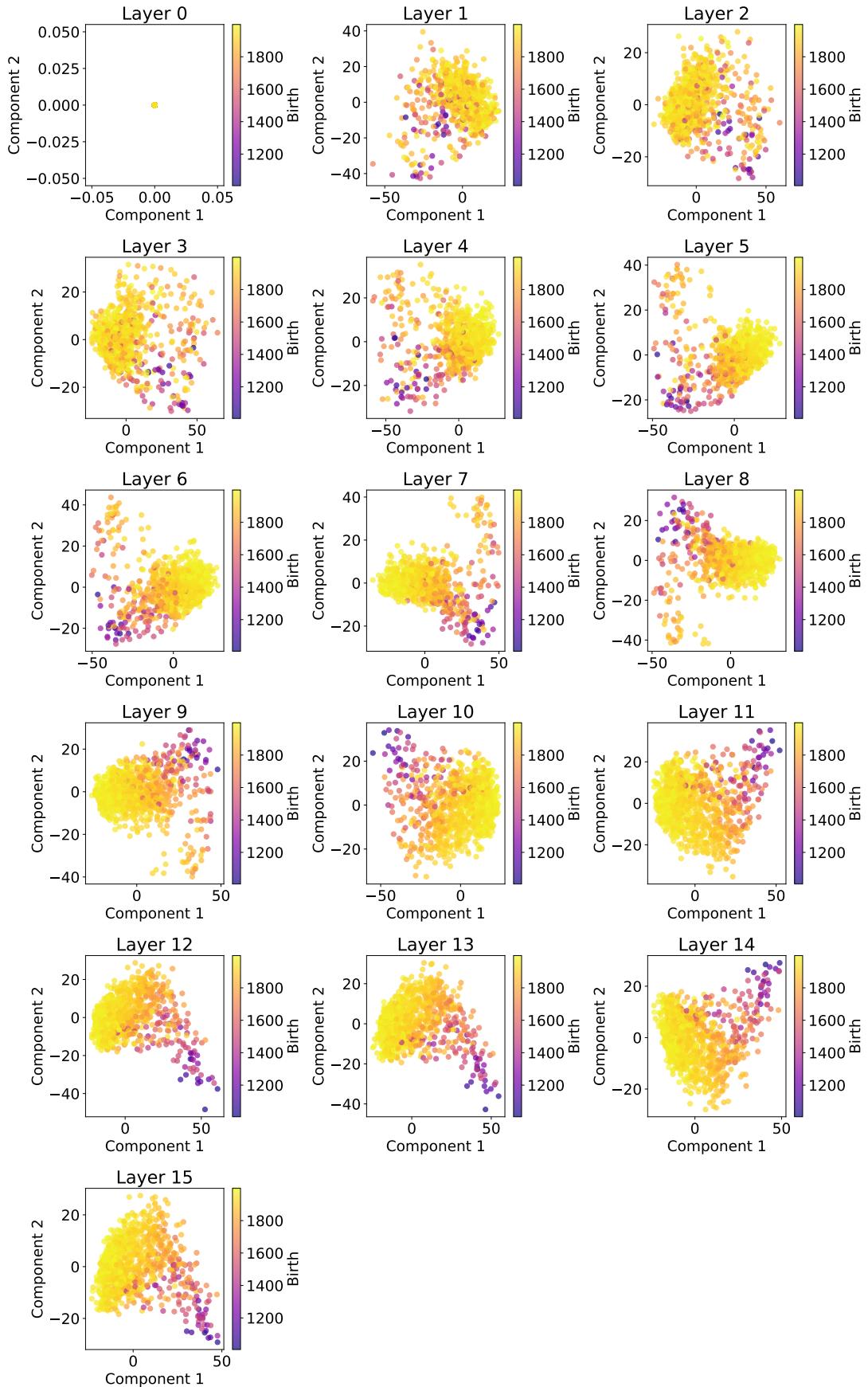


Figure 12: Two components PLS model trained on Llama-3.2-1B instruct model activations to predict entities' birth year.

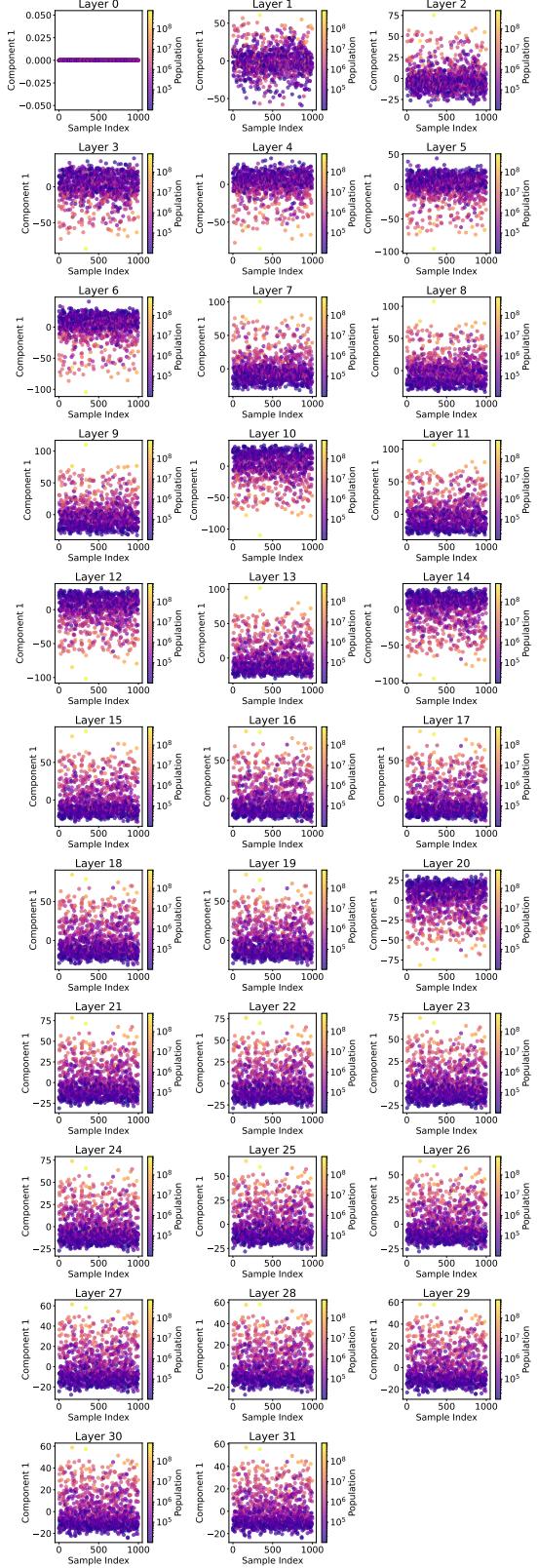


Figure 13: One component PLS model trained on Llama-3.1-8B instruct model activations to predict entities’ population size.

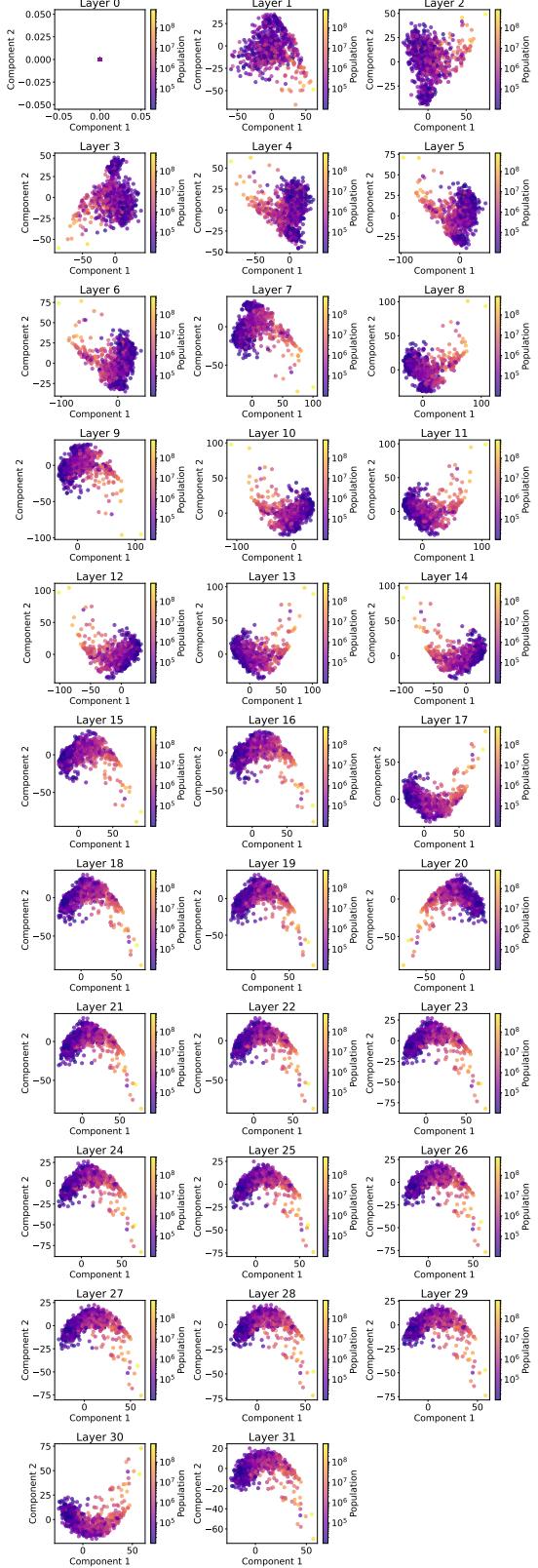


Figure 14: Two components PLS model trained on Llama-3.1-8B instruct model activations to predict entities’ population size.

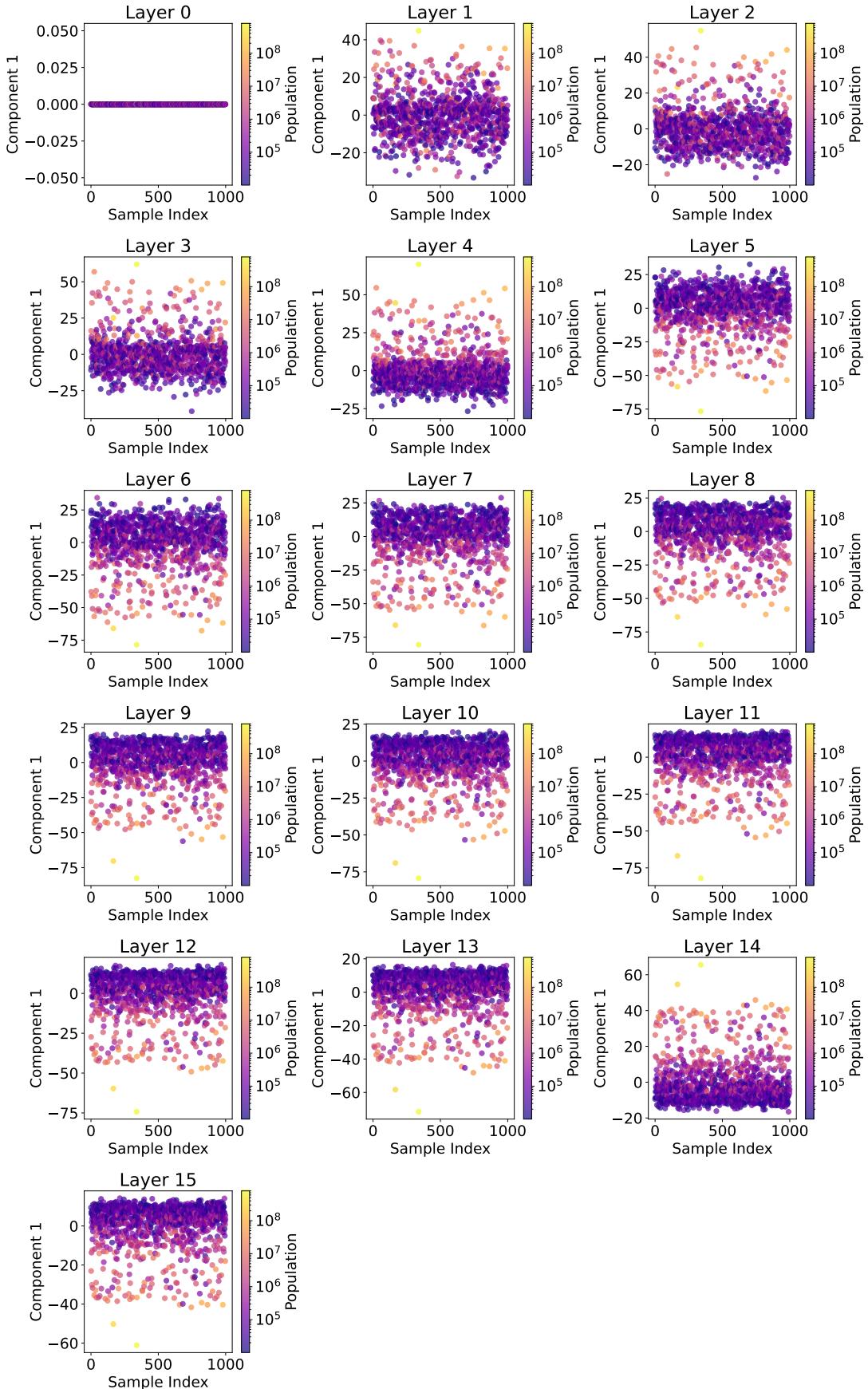


Figure 15: One component PLS model trained on Llama-3.2-1B instruct model activations to predict entities' population size.

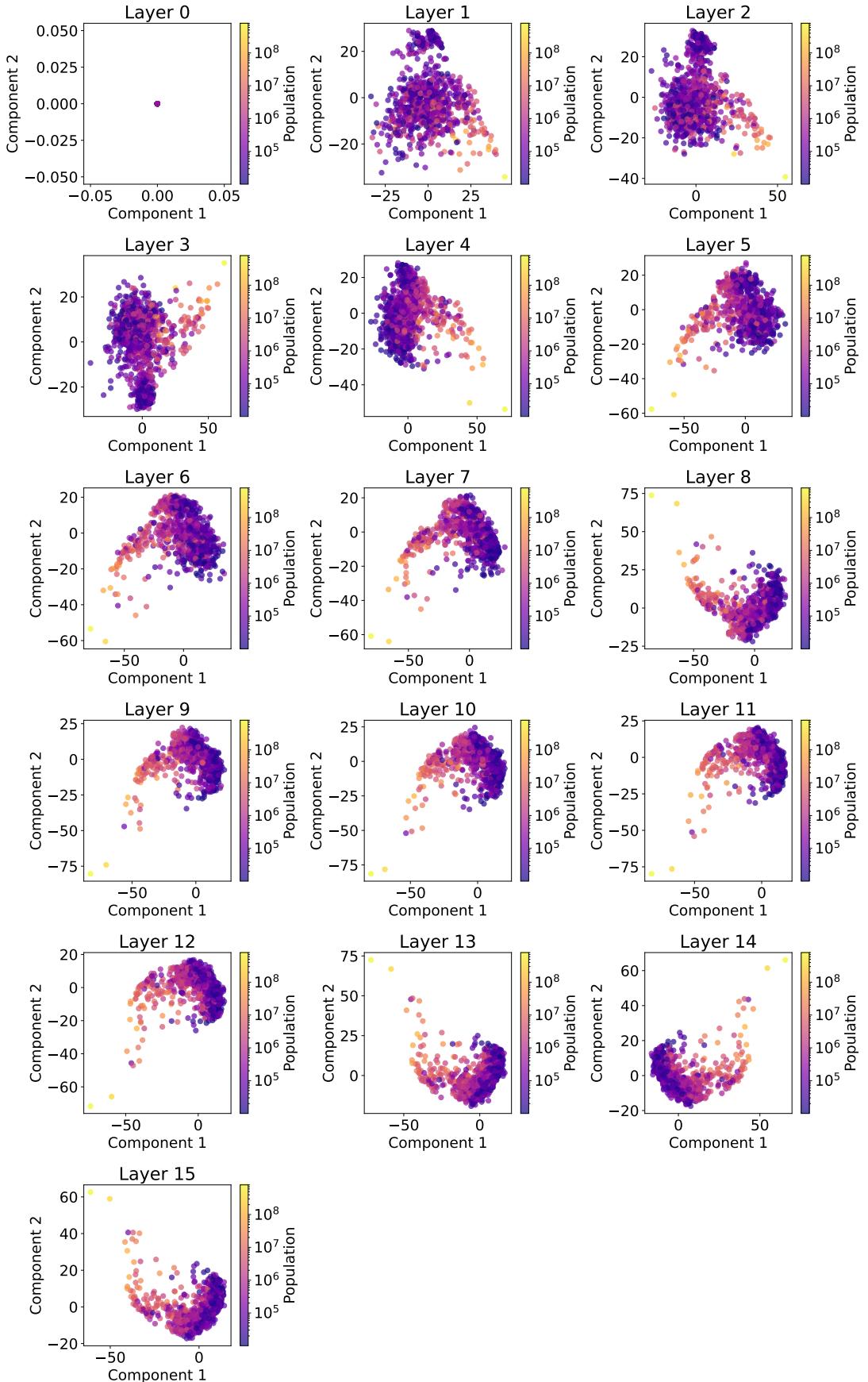


Figure 16: Two component PLS model trained on Llama-3.2-1B instruct model activations to predict entities' population size.