Language Models Do Not Embed Numbers Continuously

Alex O. Davies[1,2], **Roussel Nzoyem**[1,2], **Nirav Ajmeri**[2], **Telmo M. Silva Filho**[2]

[1]Primary Author
[2]University of Bristol, UK
alexander.davies@bristol.ac.uk, rd.nzoyemngueguin@bristol.ac.uk

Abstract

Recent research has extensively studied how large language models manipulate integers in specific arithmetic tasks, and on a more fundamental level, how they represent numeric values. These previous works have found that language model embeddings can be used to reconstruct the original values, however, they do not evaluate whether language models actually model continuous values *as* continuous. Using expected properties of the embedding space, including linear reconstruction and principal component analysis, we show that language models not only represent numeric spaces as non-continuous but also introduce significant noise. Using models from three major providers (OpenAI, Google Gemini and Voyage AI), we show that while reconstruction is possible with high fidelity ($R^2 \geq 0.95$), principal components only explain a minor share of variation within the embedding space. This indicates that many components within the embedding space are orthogonal to the simple numeric input space. Further, both linear reconstruction and explained variance suffer with increasing decimal precision, despite the ordinal nature of the input space being fundamentally unchanged. The findings of this work therefore have implications for the many areas where embedding models are used, in-particular where high numerical precision, large magnitudes or mixed-sign values are common.

Introduction

Large Language Models (LLMs), trained on next token prediction over internet-wide data, demonstrate extraordinary emergent abilities to manipulate numbers and perform arithmetic operations beyond their training date. For this reason, they are increasingly deployed in complex safety-critical scenarios requiring complex mathematical reasoning, such as accounting (Yoo 2024), medical calculations (Khandekar et al. 2024), radiotherapy planning (Wang et al. 2025), to name but a few.

These deployments pose serious safety concerns, emphasising the need to investigate how LLMs represent numbers. One problem which plagues both the expressivity and computational efficiency of LLMs is the long-range dependency (Vaswani et al. 2017; Gu and Dao 2023). It is commonly understood that LLMs perform better which shorter prompts, or focus on a specific parts of the prompt when it is too long

(Hengle et al. 2024). That said, when inserting decimal numbers in LLM prompts, users tend to include arbitrary number of decimal places, thereby prolonging the size of length of the prompt. Knowing how well LLMs represent numbers based on their precision stands to empower both users and practitioners.

The same models are also applied in scientific domains. In some works embedding models are used as-is; Peikos, Kasela, and Pasi (2024) and Amugongo et al. (2025) use multiple models for medical document retrieval, both applications that require a complex understanding of numerical values. Other works fine-tune over domain-specific data before using a model's embeddings, such as Lin et al. (2024) for geoscience applications and Choudhary (2024) for material property prediction and retrieval. Again, these works assume that LLMs are capable of usefully understanding continuous numerical values, despite significant differences in properties or semantic meaning across a range of numerical values. We provide examples from different scientific fields, with visualisations in Figure 1:

Example 1 (Climate Science). Atmospheric aerosol concentrations can range from 10^{-12}kg m^{-3} in exceptionally clean air (e.g. the Arctic) to 10^{-3}kg m^{-3} in urban areas. A scientist querying for *"Black carbon concentrations around 2.847 × 10^{-9}kg m^{-3}"* might receive "2.847 × 10^{-6} kg m^{-3}" ranked as highly similar simply because both strings share the mantissa 2.847.

Example 2 (Drug Discovery). Inhibitor potencies span many orders of magnitude: picomolar compounds (10^{-12} M) are ultra-tight binders suitable for therapeutic development, while millimolar compounds ($10{-3}$ M) bind so weakly they're considered inactive. A medicinal chemist searching for *"IC50 values near 0.0234 μM"* might retrieve compounds at 2.34 nM or 2.34 μM as similar matches.

There are also examples of where users would expect a more complex – and not strictly linear – interpretation of continuous values:

Example 3 (Astronomy). Stellar velocities within a galactic disk might range from -500 kms^{-1} (stars moving toward the observer) to +500 km^{-1} (stars moving away), with precision to 0.001 km^{-1} required to detect exoplanets via Doppler wobble. An astronomer querying for "velocity -12.847 km^{-1}" might not retrieve "+12.847 km^{-1}" as sim-
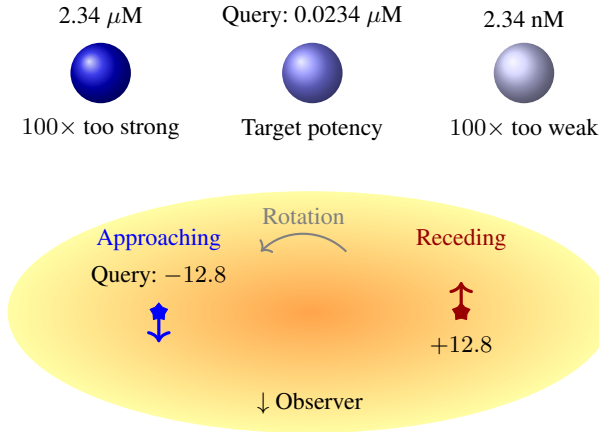
Figure 1: Visualisations of our examples for LLM embeddings in scientific knowledge applications. **Top:** In material science, the magnitude of a concentration is crucial, but repeated mantissas in numerical representations could cause incorrect retrieval. **Bottom:** In astronomy, a negatively signed value may not indicate it is semantically *opposite* to its positive counterpart, such as in measuring velocities within galactic disks.

ilar, despite the physical equivalency on opposite sides of the galactic disk, as the magnitude is opposite.

The behaviours these models exhibit in embedding numerical values is therefore crucial before application. A model which embeds strictly linearly is useful in the climate and material science examples, but would require more complex treatment where magnitude does not necessarily imply dis-similarity.

Since other recent works have highlighted representational space as critical for arithmetic (Maltoni and Ferrara 2024; Zhu, Dai, and Sui 2024), a plethora of methods have attempted to understand the semantics of the LLM embedding space. The current literature is heavily focused on the representation and processing of integers within specific arithmetic or reasoning tasks (Levy and Geva 2024; Kantamneni and Tegmark 2025). Some studies have attempted to reframe numerical representations to improve LLM performance (Schwartz et al. 2024; Zhang-Li et al. 2024), while others have brought attention to the connections to human cognition (Shah et al. 2023; AlquBoj et al. 2025). While this has yielded fascinating insights into the complex, often non-linear geometry of numerical representations, it leaves a more fundamental question unanswered: *how well do embedding models encode the basic semantic value of continuous real numbers across varying scales and precisions?*

Our work addresses this gap by providing a **general** and **lightweight** framework for evaluating the semantic fidelity of numerical embeddings. Rather than decoding a specific geometric structure (e.g., a helix or circle (Kantamneni and Tegmark 2025)), we propose a set of metrics (namely linear R^2, PCA correlation and explained variance) that directly quantify how well an embedding captures the one-dimensional nature of a scalar value. Using these metrics we

provide a critical insight: the complex, multi-faceted representations identified in other studies (string-entanglements, periodic features) manifest as quantifiable "noise" in the embedding space. Our methodology thus offers a scalable and task-agnostic tool to measure the purity and robustness of any model's ability to represent the simple concept of numerical magnitude[1].

Specifically, our contributions are as follows:

(1) A general and lightweight framework to evaluate the fidelity of continuous embeddings, addressing a critical gap in the literature in a model-agnostic manner.

(2) Task-independent metrics (linear R^2, PCA correlation, and explained variance) which demonstratively quantify how well an embedding captures the ordinal, one-dimensional nature of the scalar values. These quantify the fidelity of numerical representations, enabling practitioners to better understand model limitations and optimize prompt design.

(3) A breadth of experiment to validate our framework and the proposed metrics. We go beyond the scope of the current literature by considering positive decimals, mixed sign decimals, and mixed sign integers.

Related Work

A central challenge for Large Language Models (LLMs) is their inconsistent and often fragile ability to perform numerical reasoning. Recent research has tackled this from multiple angles: improving arithmetic performance through novel processing techniques, probing the internal geometry of numerical representations, and drawing parallels between model and human numerical cognition.

Internal Representations and Processing Methods The current theme in the literature is that LLMs introduce complex patterns into embeddings for simple 1D scalar values (Zhu, Dai, and Sui 2024). Probing experiments reveal highly complex internal structures. For instance, numbers appear to be encoded using per-digit circular representations in base 10 (Levy and Geva 2024), which helps explain why model errors are often digit-based rather than value-based. For specific operations like addition, models have been shown to develop even more intricate structures, representing numbers as a generalized helix and manipulating them with a trigonometric "Clock" algorithm (Kantamneni and Tegmark 2025). While limited to whole number representations, Kantamneni and Tegmark (2025) show that LLMs contain the ability to abstract away the continuous space. Zhou et al. (2024) identify Fourier features with varying levels of periodicity within these representations, which are later used for arithmetic operations. This internal complexity is further compounded by a fundamental ambiguity: LLM representations are often an entanglement of a number's value and its string-like properties, where similarity is influenced by various metrics of distance (Marjieh et al. 2025).

---

[1] Our codebase will be made available, via GitHub, in the camera-ready version.

Reformatting Numerical Representations: To improve performance in light of these complex representations, researchers have focused on reformatting inputs to align with computational logic. NumeroLogic (Schwartz et al. 2024), for example, prefixes numbers with their digit count to provide essential place-value context upfront. In a similar vein, Little-Endian Fine-Tuning (LEFT) (Zhang-Li et al. 2024) reverses the order of digits to mimic human-like computation (least significant digit first), dramatically improving efficiency and accuracy in arithmetic tasks. Neither work, unlike ours, investigates the role of number magnitude and precision. The need for these techniques is underscored by comprehensive benchmarks which reveal that modern LLMs still fail at a wide range of basic numerical tasks. For example, the NUPA Test (Yang et al. 2024) shows broad failures beyond simple addition, while Tang et al. (2025) highlight significant error rates in the seemingly straightforward task of numerical translation, especially when dealing with large units across languages.

Experiments

Consider a real scalar number $x \in \mathbb{R}^1$, and an embedding model $f(x) \rightarrow \hat{x} \in \mathbb{R}^d$ for d the dimensionality of the model's embedding space. Scalars x are in a set of $X = \{x_1, x_2, \ldots\}$. Further, consider that $x \in X$ has a given number of integer and decimal places a and b;

$$\underbrace{1234}_{a=4} \cdot \underbrace{567}_{b=3}$$

In this work we evaluate how accurately embedding models encode numbers x with respect to these precisions a, b and sign of the number. A diagram of our experimental framework is presented in Figure 2. We can expect that, as in prior work (Zhu, Dai, and Sui 2024; Levy and Geva 2024), a number x can be reproduced by a linear model over the embedding $\text{lin}(\hat{X}) \rightarrow X'$; perfect reconstruction is $\text{lin}(\hat{X}) = X$. More precisely, we expect that there is very good correlation between predicted values from this linear model X' and the original scalars X.

$$\text{corr}(X', X) \simeq 1 \tag{1}$$

Further, we know that there is only one component of variation, with $\text{Rank}(X) = 1$. As a result we expect that strong embeddings of numbers, with 'understanding' of numeric spaces, would similarly have only one component of variation, $\text{Rank}(\hat{X}) \simeq 1$. As a result, Principal Component Analysis (PCA) over the embedded scalars $\hat{X}$ should have an explained variance ratio in the first component of approximately 1:

$$\text{VR} = \frac{\lambda_0}{\sum_0^d \lambda_i} \simeq 1 \tag{2}$$

with λ_i the eigenvalues of the covariance matrix. We can also expect that the primary direction of variation in the embedding space is in-line with the original set of scalars X:

$$\text{corr}(\text{PCA}_0, X) \simeq 1. \tag{3}$$

For a perfect encoder, and continuous internal representations, we expect the relationships in Equations 1,2 and 3 to hold true regardless of integer and decimal places a, b or the signs of the numbers.
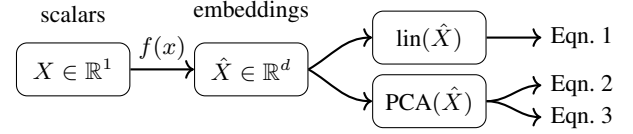

Figure 2: Framework for measuring numerical embedding quality. Scalars are embedded into high-dimensional space and evaluated using linear reconstruction and PCA to quantify preservation of numerical structure through three complementary metrics, defined in Equations (1,2,3).

Datasets We produce three datasets of scalars X. The first is positive decimals $x \in [0, 1]$ with $a = 1$ and with the precision of the decimals iteratively increased from $a = 1$ to $a = 20$. The second is mixed sign decimals $x \in [-1, 1]$ with the same precisions as the positive decimals. The third is mixed-sign scalars of varied integer and zero decimal places $b = 0, a \in [0, 20]$. The range of this dataset varies according to a, $-10^{(a)} < x < 10^{(a)}$. Across the three datasets individual scalars are sampled randomly, up to 500 samples per dataset. We adopt a 5-fold split to produce error margins.

Embedding Models In our evaluation of LLMs as embedding models for numeric data, we use models which are specifically targetted at embedding applications. We use all of the embedding models available from three public providers:

Gemini [2]Provided by Google, we use the `gemini-embedding-001` model with the default (largest) embedding dimension. Gemini embedding models are initialised from the Gemini LLM, with two rounds of contrastive training to produce the embedding model (Lee et al. 2025). Gemini embedding models are reported to out-perform competitors on benchmarks.

OpenAI [3] we use the three latest models provided by OpenAI, namely `text-embedding-3-large`, a condensed version `text-embedding-3-large` and the older`text-embedding-ada-002`. No companion paper is published, but public documentation and releases suggests that OpenAI's embedding models are based on the GPT-4 family, and are contrastively fine-tuned for embeddings (OpenAI 2025). OpenAI models are reported to out-perform competitors on benchmarks.

VoyageAI [4], provided by MongoDB, provides several text embedding models. We evaluate their non-specialist models, namely the 3.5 series (default, `lite` and
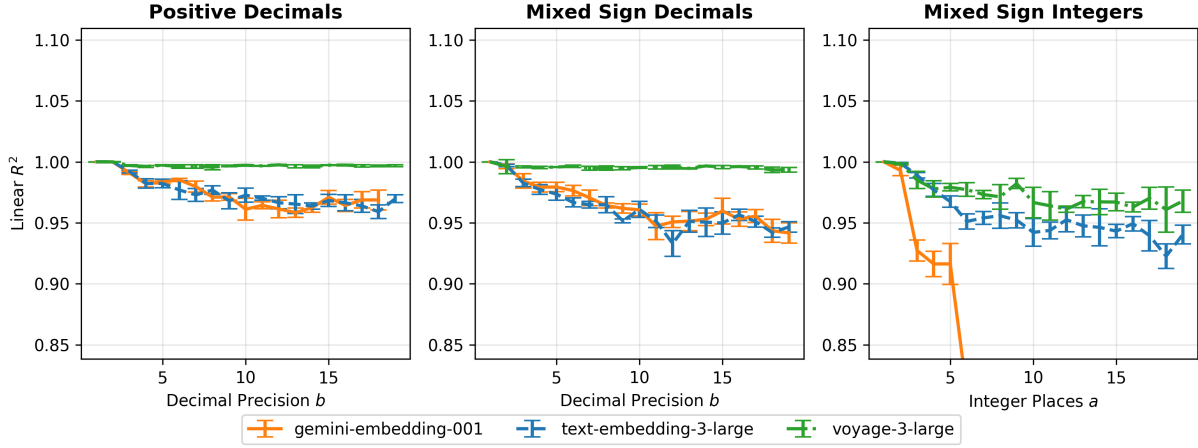
---

[2]https://ai.google.dev/gemini-api/docs/embeddings
[3]https://platform.openai.com/docs/guides/embeddings
[4]https://docs.voyageai.com/docs/embeddings

Figure 3: Decimal precision for each dataset plotted against the R^2 score of the linear model reconstructing the original scalars X from their embedded counterparts $\hat{X}$.
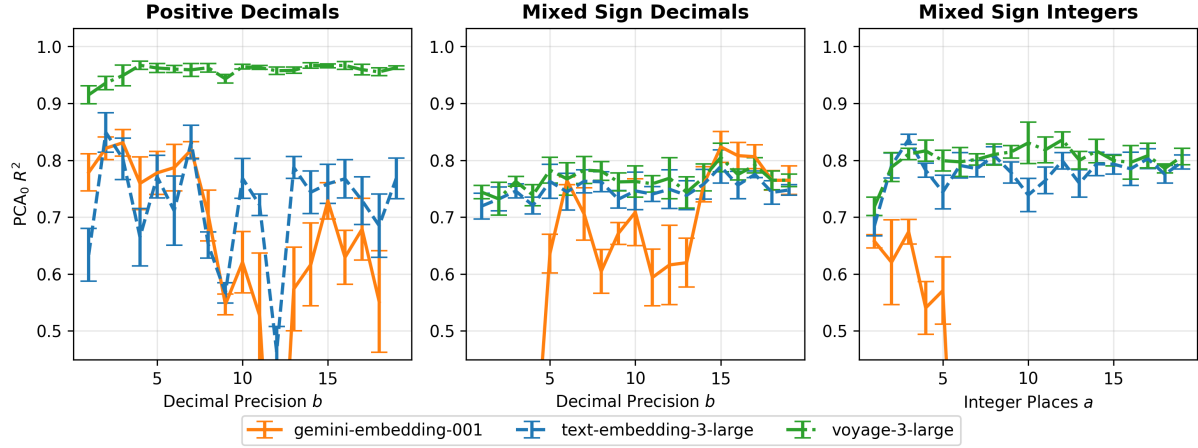

Figure 4: Decimal precision for each dataset plotted against the R^2 of the first component of a PCA projection of the embedded samples $\hat{X}$ against their original counterparts $\hat{X}$.
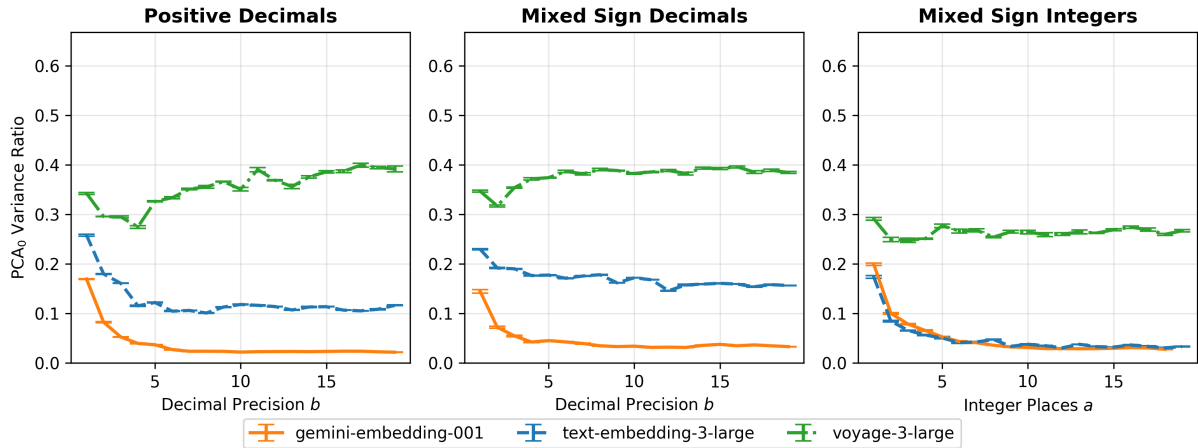

Figure 5: Decimal precision for each dataset plotted against the explained variance ratio of the first component of a PCA projection of the embedded samples $\hat{X}$.

large). No companion paper is published, with public documentation suggesting pure contrastive training for these embedding models (VoyageAI 2025). VoyageAI models are reported to out-perform competitors on benchmarks.

Quantitative results for each dataset and metric can be found in Table 1 for positive decimals, in Table 2 for mixed-sign decimals, and in Table 3 for high-magnitude mixed-sign integers.

Linear Reconstruction

Here we evaluate the relationship in Equation 1, that is, a linear model can perfectly reconstruct samples X from the corresponding samples in the embedding space $\hat{X}$. Figure 3 shows linear R^2 scores plotted against size a and b for each dataset.

On positive decimals performance for all models degrades as the precision (b) increases, though most models maintain $\text{corr}(X, X') \geq 0.95$, indicating that the original samples can be well-constructed from the embedding space even at very high precisions. Introducing mixed-sign decimals degrades performance for all models, most notably for condensed models. OpenAI models in-particular degrade in performance far more with decimal precision when negative decimals are included. Allowing integer places, and larger magnitudes, leads to greater degradation from all models. In-particular, Google's Gemini-based model drops to at-best medium correlation with precision beyond $a = b = 7$.

Overall we echo the findings of prior work that LLM embeddings of numbers can broadly be used to reconstruct those numbers. However, when numbers are allowed to range in sign and magnitude, such reconstruction suffers. Notably, for true 'understanding' of the simple numeric space our models are encoding, there should be little to no variation in these correlations with integer or decimal places.

PCA Correlation

Next, to evaluate the preservation of Equation 3, we measure the correlation of the first PCA component PCA_0 over the embedded datasets $\hat{X}$ against the original scalars they represent. This measures, in effect, whether the embedding correctly encodes the direction of the input samples. Perfect encoders would lead to perfect correlation between PCA_0 and X. We visualise these results in Figure 4.

Performance on positive decimals is highly volatile, with most models dipping with increasing precision into low correlations. Only voyage-3-large maintains high correlations against increasing decimal precision, though all models at one decimal place are at least fairly well correlated with the original samples. On both the larger magnitude and mixed sign decimals the Gemini model again performs poorly, though on the mixed sign dataset it recovers some performance with increasing precision. Notably all other models perform comparably on the mixed sign decimal dataset, with little overall degradation as precision increases, and at roughly the same correlation values as on the positive decimals dataset. We discuss these results in more detail in the Discussion, but overall we have shown that the

principle component of the embedded samples contains at least most of the information necessary to reconstruct the ordering of X.
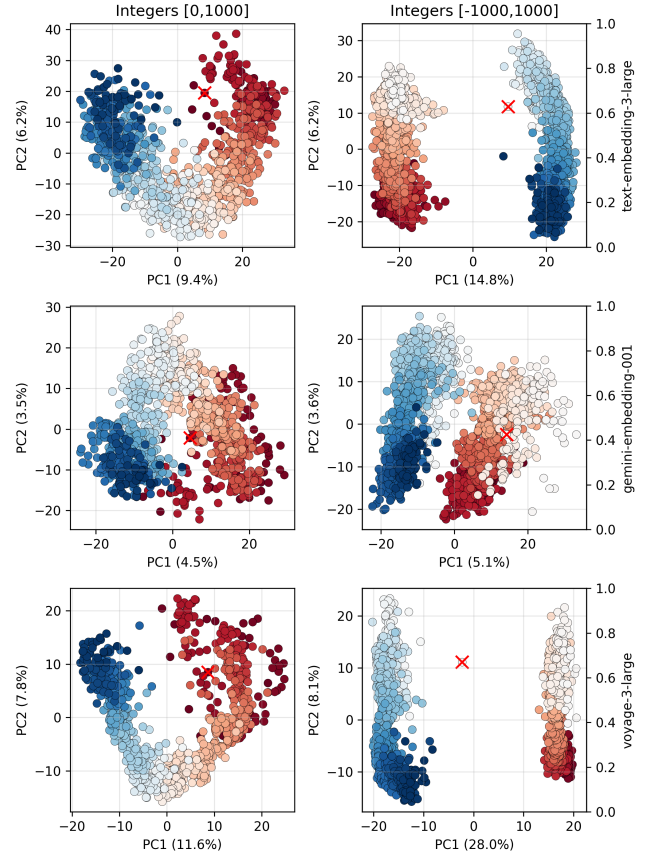
PCA Explained Variance

Figure 6: Visualisations of the first two principal components of embeddings of the integers $x \in [0, 1000]$ (left) and $x \in [-1000, 1000]$ (right) for the main OpenAI, Voyage and Gemini models. '$\times$' symbols mark $x = 0$.

Finally we evaluate the preservation of Equation 2, measuring the relative amount of variance explained by the first principle component PCA_0. We expect, given the one-dimensional and uniform nature of the input samples X, that PCA_0 explains *all* of the variance in the embedded samples $\hat{X}$. We visualise the explained variance ratio for each model and dataset against increasing precision in Figure 5.

On positive decimals all models show the same exponential-like explained variance decrease with increasing decimal precision, with at most 40% of variance explained at one decimal place. Results are more spread on the mixed sign dataset, with explained variances increasing for most models, with the same pattern on the larger magnitude integer and decimal dataset. Voyage models overall explain more variance in their first principle component, and actually increase in this explained variance as precision increases. Models from OpenAI and Google simply decrease

Table 1: Metrics for linear models and principal components over a dataset of positive numbers of varying decimal places.

Model	Provider	Linear R^2		PCA R^2		PCA Variance	
		Min	Max	Min	Max	Min	Max
gemini-embedding-001	Google	0.96 ± 0.00	1.00 ± 0.00	0.04 ± 0.04	0.83 ± 0.02	0.03 ± 0.00	0.20 ± 0.00
text-embedding-3-large	OpenAI	0.96 ± 0.01	1.00 ± 0.00	0.46 ± 0.04	0.85 ± 0.03	0.03 ± 0.00	0.17 ± 0.00
text-embedding-3-small	OpenAI	0.96 ± 0.01	1.00 ± 0.00	0.64 ± 0.04	0.87 ± 0.03	0.04 ± 0.00	0.19 ± 0.00
text-embedding-ada-002	OpenAI	0.95 ± 0.01	1.00 ± 0.00	0.25 ± 0.03	0.85 ± 0.02	0.04 ± 0.00	0.17 ± 0.00
voyage-3-large	Voyage	1.00 ± 0.00	1.00 ± 0.00	0.92 ± 0.02	0.97 ± 0.01	0.25 ± 0.00	0.29 ± 0.00
voyage-3.5	Voyage	0.97 ± 0.01	1.00 ± 0.00	0.02 ± 0.02	0.93 ± 0.01	0.14 ± 0.00	0.36 ± 0.00
voyage-3.5-lite	Voyage	0.97 ± 0.01	1.00 ± 0.00	0.64 ± 0.02	0.86 ± 0.01	0.18 ± 0.00	0.33 ± 0.00

Table 2: Metrics for linear models and principal components over a dataset of numbers of varying decimal places and mixed signs.

Model	Provider	Linear R^2		PCA R^2		PCA Variance	
		Min	Max	Min	Max	Min	Max
gemini-embedding-001	Google	0.94 ± 0.01	1.00 ± 0.00	0.01 ± 0.01	0.82 ± 0.03	0.03 ± 0.00	0.14 ± 0.00
text-embedding-3-large	OpenAI	0.93 ± 0.01	1.00 ± 0.00	0.72 ± 0.02	0.79 ± 0.03	0.15 ± 0.00	0.23 ± 0.00
text-embedding-3-small	OpenAI	0.87 ± 0.01	1.00 ± 0.00	0.71 ± 0.02	0.78 ± 0.03	0.18 ± 0.00	0.30 ± 0.00
text-embedding-ada-002	OpenAI	0.85 ± 0.03	1.00 ± 0.00	0.69 ± 0.04	0.77 ± 0.02	0.13 ± 0.00	0.21 ± 0.00
voyage-3-large	Voyage	0.99 ± 0.00	1.00 ± 0.00	0.73 ± 0.03	0.80 ± 0.02	0.32 ± 0.00	0.40 ± 0.00
voyage-3.5	Voyage	0.96 ± 0.01	1.00 ± 0.00	0.73 ± 0.02	0.79 ± 0.02	0.38 ± 0.00	0.45 ± 0.00
voyage-3.5-lite	Voyage	0.91 ± 0.11	1.00 ± 0.00	0.75 ± 0.02	0.80 ± 0.02	0.37 ± 0.00	0.55 ± 0.00

in performance with increasing precision, with the Gemini model explaining the least variance in its first principle component.

Discussion

Our results reveal a complex picture of how LLM embedding models encode numerical information. First, we demonstrate that embeddings can indeed be used to reconstruct numbers with reasonable fidelity. The linear reconstruction experiments show that most models maintain R^2 scores above 0.95 for simpler numerical datasets, confirming that numerical information is preserved in the embedding space. This finding supports prior work suggesting that language models possess some inherent understanding of numerical relationships.

Second, the first principal component of the embedded representations correlates meaningfully with input precision across most models and datasets. This correlation indicates that the primary axis of variation in the embedding space aligns with the numerical ordering of the input scalars, suggesting that models do capture the fundamental ordinal structure of numbers.

However, our third finding reveals a significant limitation: the explained variance by the first principal component remains consistently low across all models, typically below 40% even for the simplest datasets. This low explained variance is particularly concerning when considered alongside our fourth observation. Given that the first component does correlate well with the original numbers, the low explained variance implies that the embedding space contains substantial additional variation that is not present in the original one-dimensional numerical input.

In Figure 6 we visualise the first two principal components of these principal components for the 'flagship' model from each provider over the complete sets of integers $x_+ \in [0, 1000]$ and $x_\pm \in [-1000, 1000]$. x_+ shows that while the first principal component does broadly encode the rank of the original data, the second principal component introduces clear trends unrelated to the original values. Further, all three models on $x_\pm$ have the first principal component encoding effectively only the sign of the data, despite the fundamentally continuous nature of the original values. The second component then broadly encodes magnitude, although for the gemini model there is significant 'bleed' between the clusters for positive and negative values. In the Appendix we reproduce Figure 6 for increasing number magnitude, and observe that for large magnitudes neither the first nor second principal components represent number magnitude.

This excess variation suggests that embedding models introduce considerable noise into their numerical representations. The high-dimensional embedding spaces capture not only the intended numerical information but also artifacts from the models' pretraining on diverse text corpora. These artifacts manifest as spurious dimensions of variation that obscure the underlying numerical structure.

Implications

Our findings have several practical implications for applications utilizing LLM embeddings for numerical data. First, numerical understanding appears to degrade significantly with increasing precision, suggesting that simply rounding numbers to fewer decimal places may improve performance in downstream tasks. This finding is particularly relevant for applications requiring numerical reasoning or similarity

Table 3: Metrics for linear models and principal components over a dataset of numbers of varying integer places with mixed signs.

Model	Provider	Linear R^2		PCA R^2		PCA Variance	
		Min	Max	Min	Max	Min	Max
gemini-embedding-001	Google	0.48 ± 0.08	0.99 ± 0.01	-0.03 ± 0.02	0.72 ± 0.02	0.02 ± 0.00	0.10 ± 0.00
text-embedding-3-large	OpenAI	0.91 ± 0.02	1.00 ± 0.00	0.69 ± 0.03	0.84 ± 0.02	0.10 ± 0.00	0.19 ± 0.00
text-embedding-3-small	OpenAI	0.89 ± 0.01	1.00 ± 0.00	0.41 ± 0.07	0.82 ± 0.01	0.07 ± 0.00	0.20 ± 0.00
text-embedding-ada-002	OpenAI	0.90 ± 0.01	1.00 ± 0.00	0.59 ± 0.03	0.79 ± 0.01	0.09 ± 0.00	0.17 ± 0.00
voyage-3-large	Voyage	0.95 ± 0.02	1.00 ± 0.01	0.73 ± 0.01	0.83 ± 0.02	0.28 ± 0.00	0.44 ± 0.00
voyage-3.5	Voyage	0.93 ± 0.01	1.00 ± 0.00	0.67 ± 0.02	0.85 ± 0.02	0.29 ± 0.00	0.38 ± 0.00
voyage-3.5-lite	Voyage	0.90 ± 0.06	1.00 ± 0.00	0.68 ± 0.02	0.87 ± 0.02	0.26 ± 0.00	0.42 ± 0.00

computation over quantitative data.

Second, the signs of numbers have substantial impact on embedding quality across all tested models. The principal component of variation for all models, with mixed sign values, comes to represent only the sign of the numbers (see Figure 6). The introduction of negative values consistently degrades performance metrics, indicating that models struggle with the concept of negative numbers more than might be expected. This limitation suggests caution when using embeddings for datasets containing both positive and negative values.

Third, embedding models introduce systematic noise into numerical representations, with those based on large language models being particularly prone to this issue. The low explained variance ratios demonstrate that much of the embedding space is devoted to capturing information orthogonal to the numerical content itself. This noise may interfere with applications requiring precise numerical relationships, such as mathematical reasoning or quantitative analysis tasks.

Conclusion

We have conducted a comprehensive evaluation of numerical precision in LLM embedding models, examining how well these systems encode scalar values across different ranges and precisions. Our analysis reveals that while embedding models can preserve numerical information sufficiently for linear reconstruction, they introduce substantial noise that limits their effectiveness for precise numerical applications.

The key finding is that embedding models exhibit a fundamental trade-off between preserving numerical information and introducing extraneous variation. While the primary component of variation in embeddings does correlate with numerical values, the majority of the embedding space encodes information unrelated to the numerical content. This suggests that current embedding models, despite their success in many natural language processing tasks, may not be optimal for applications requiring precise numerical understanding.

Future work should focus on developing embedding architectures specifically designed for numerical data, potentially through specialized pretraining objectives or architectural modifications that better isolate numerical information from other sources of variation. Additionally, investigating techniques for denoising numerical embeddings or identifying the most relevant dimensions for numerical tasks could improve the practical utility of existing models for quantitative applications.

References

AlquBoj, H.; AlQuabeh, H.; Bojkovic, V.; Hiraoka, T.; El-Shangiti, A. O.; Nwadike, M.; and Inui, K. 2025. Number Representations in LLMs: A Computational Parallel to Human Perception. *arXiv:2502.16147*.

Amugongo, L. M.; Mascheroni, P.; Brooks, S.; Doering, S.; and Seidel, J. 2025. Retrieval augmented generation for large language models in healthcare: A systematic review. *PLOS Digital Health*, 4(6): 1–33.

Choudhary, K. 2024. AtomGPT: Atomistic Generative Pretrained Transformer for Forward and Inverse Materials Design. *The Journal of Physical Chemistry Letters*, 15(27): 6909–6917.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv:2312.00752*.

Hengle, A.; Bajpai, P.; Dan, S.; and Chakraborty, T. 2024. Multilingual needle in a haystack: Investigating long-context behavior of multilingual large language models. *arXiv:2408.10151*.

Kantamneni, S.; and Tegmark, M. 2025. Language models use trigonometry to do addition. *arXiv:2502.00873*.

Khandekar, N.; Jin, Q.; Xiong, G.; Dunn, S.; Applebaum, S.; Anwar, Z.; Sarfo-Gyamfi, M.; Safranek, C.; Anwar, A.; Zhang, A.; et al. 2024. Medcalc-bench: Evaluating large language models for medical calculations. *Advances in Neural Information Processing Systems*, 37: 84730–84745.

Lee, J.; Chen, F.; Dua, S.; Cer, D.; Shanbhogue, M.; Naim, I.; Ábrego, G. H.; Li, Z.; Chen, K.; Vera, H. S.; Ren, X.; Zhang, S.; Salz, D.; Boratko, M.; Han, J.; Chen, B.; Huang, S.; Rao, V.; Suganthan, P.; Han, F.; Doumanoglou, A.; Gupta, N.; Moiseev, F.; Yip, C.; Jain, A.; Baumgartner, S.; Shahi, S.; Gomez, F. P.; Mariserla, S.; Choi, M.; Shah, P.; Goenka, S.; Chen, K.; Xia, Y.; Chen, K.; Duddu, S. M. K.; Chen, Y.; Walker, T.; Zhou, W.; Ghiya, R.; Gleicher, Z.; Gill, K.; Dong, Z.; Seyedhosseini, M.; Sung, Y.; Hoffmann, R.; and Duerig, T. 2025. Gemini Embedding: Generalizable Embeddings from Gemini. arXiv:2503.07891.

Levy, A. A.; and Geva, M. 2024. Language models encode numbers using digit representations in base 10. *arXiv:2410.11781*.

Lin, Z.; Deng, C.; Zhou, L.; Zhang, T.; Xu, Y.; Xu, Y.; He, Z.; Shi, Y.; Dai, B.; Song, Y.; Zeng, B.; Chen, Q.; Miao, Y.; Xue, B.; Wang, S.; Fu, L.; Zhang, W.; He, J.; Zhu, Y.; Wang, X.; and Zhou, C. 2024. GeoGalactica: A Scientific Large Language Model in Geoscience. *arXiv*.

Maltoni, D.; and Ferrara, M. 2024. Arithmetic with language models: From memorization to computation. *Neural Networks*, 179: 106550.

Marjieh, R.; Veselovsky, V.; Griffiths, T. L.; and Sucholutsky, I. 2025. What is a Number, That a Large Language Model May Know It? *arXiv preprint arXiv:2502.01540*.

OpenAI. 2025. New embedding models and API updates. https://openai.com/index/new-embedding-models-and-api-updates/. Accessed: 10/2025.

Peikos, G.; Kasela, P.; and Pasi, G. 2024. Leveraging Large Language Models for Medical Information Extraction and Query Generation. In *2024 IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, 367–372.

Schwartz, E.; Choshen, L.; Shtok, J.; Doveh, S.; Karlinsky, L.; and Arbelle, A. 2024. NumeroLogic: Number Encoding for Enhanced LLMs' Numerical Reasoning. *arXiv:2404.00459*.

Shah, R.; Marupudi, V.; Koenen, R.; Bhardwaj, K.; and Varma, S. 2023. Numeric magnitude comparison effects in large language models. In *Findings of ACL 2023*, 6147–6161.

Tang, W.; Yu, J.; Li, Y.; Zhao, Y.; Zhang, W.; Feng, W.; Zhang, M.; and Yang, H. 2025. Investigating Numerical Translation with Large Language Models. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

VoyageAI. 2025. voyage-3-large: the new state-of-the-art general-purpose embedding model. https://blog.voyageai.com/2025/01/07/voyage-3-large/. Accessed: 10/2025.

Wang, Q.; Wang, Z.; Li, M.; Ni, X.; Tan, R.; Zhang, W.; Wubulaishan, M.; Wang, W.; Yuan, Z.; Zhang, Z.; et al. 2025. A feasibility study of automating radiotherapy planning with large language model agents. *Physics in Medicine & Biology*, 70(7): 075007.

Yang, H.; Hu, Y.; Kang, S.; Lin, Z.; and Zhang, M. 2024. Number cookbook: Number understanding of language models and how to improve it. *arXiv:2411.03766*.

Yoo, M. 2024. How Much Should We Trust LLM-Based Measures for Accounting and Finance Research? *SSRN*.

Zhang-Li, D.; Lin, N.; Yu, J.; Zhang, Z.; Yao, Z.; Zhang, X.; Hou, L.; Zhang, J.; and Li, J. 2024. Reverse that number! decoding order matters in arithmetic learning. *arXiv:2403.05845*.

Zhou, T.; Fu, D.; Sharan, V.; and Jia, R. 2024. Pre-trained large language models use fourier features to compute addition. *Advances in Neural Information Processing Systems*, 37: 25120–25151.

Zhu, F.; Dai, D.; and Sui, Z. 2024. Language models know the value of numbers. *arXiv:2401.03735*.