UNIVERSITÀ
DI TORINO

Università degli Studi di Torino

Corso di Laurea Magistrale in Informatica

Concrete Numeric Representations in LLM Embeddings

Tesi di Laurea

Relatore/Relatrice
Prof. Di Caro Luigi

Correlatore/Correlatrice
Dr. Torrielli Federico

Candidato/a
Gentiletti Emanuele
900831

Anno Accademico 2024/2025

Contents

Introduction

This work started with a simple premise: why are LLMs bad at math?

This is not really a hard question to answer. Most of the LLMs to date are not built with that purpose in mind, and can rely on tool calling to give good answers to quantitative and numerical questions.

There is a tremendous investment in computing resources that is directed towards arithmetic operations that make up the inner workings of LLMs, computations that the LLMs themselves aren't capable of leveraging to answer arithmetic questions. It feels like witnessing a fundamental disconnection, where the LLM is segregated from the capabilities that make its own functioning possible.

Savant syndrome is a very rare disorder. It manifests primarily in people with autism spectrum disorders (Murray, 2010) or after traumatic episodes. The people affected by it possess extraordinary qualities in certain areas, like arts, music or mathematics, while usually showing significant impairment in others. One of the possible areas in which savants may show exceptional aptitude is calculation: calendrical savants are able to instantly know the day of the week of dates far in the future. These skills are unlikely to be the product of algorithmic calculation (Cowan & Frith, 2009), so alternative hypotheses emerged.

What I propose here is that the Savant condition can be seen as a parallel to the bridging of this capabilities gap in LLMs. In particular, what is taken in consideration here is the use of concrete representations as described in (Murray, 2010), where abstract numerical concepts are transformed into "highly accessible concrete representations" that can be directly manipulated rather than computed through algorithmic steps. This reification process - the conversion of abstract concepts into concrete entities - appears to provide savants with immediate access to numerical relationships that would otherwise require complex calculations.

This is not meant necessarily to give a comprehensive explanation of the phenomenon on an empirical basis, as that would be hard to establish from the basis of current knowledge about both savant cognition and neural network representations. Rather, it serves as a conceptual framework for exploring whether similar representational advantages can be induced in artificial systems.

This idea is explored in two ways:

- by a literature review, that is meant to clarify what can function as concrete representations in this context
- by an exploration of numerical embeddings, that is meant to show whether the learned representation of current language models already tends to conform to certain geometrical objects or structures. We show that there is remarkable structure and patterns in the learned representation of current LLMs.

The Transformer architecture and vector representations

The inductive bias of Tokenization

Modern LLMs are built on the Transformer architecture (Vaswani et al., 2023), which operates by converting input text into sequences of discrete tokens that are then mapped to high-dimensional vector representations. This initial tokenization step creates an inductive bias that shapes how the model processes information (Ali et al., 2024) (Singh & Strouse, 2024), with significant implications for the application of the numerical data to arithmetical tasks.

The most used algorithm for tokenization is currently Byte-Pair Encoding, which, given a fixed vocabulary size, starts with individual characters and iteratively merges the most frequently occurring pairs of adjacent tokens until the vocabulary limit is reached. This process naturally creates longer tokens for common substrings that appear frequently in the training data. For numbers, this means that frequently occurring numerical patterns like "100", "2020", or "999" might become single tokens, while less common numbers get broken into smaller pieces. The result is an idiosyncratic and unpredictable tokenization scheme where similar numbers can be tokenized completely differently based purely on their frequency in the training corpus. While GPT-2 used to have a purely BPE tokenizer, the successive iteration of GPT and generally more recent models either tokenize digits separately (so as $'1234' \rightarrow [1, 2, 3, 4]$), or tokenize clusters of 3 digits, encompassing the integers in the range 0-999.

Most of the tokenizers right now do L2R (left-to-right) clustering, meaning that a number such as 12345 would be divided in two tokens, 123 and 45. It has been shown (Singh & Strouse, 2024) that this kind of clustering leads to a lesser arithmetic performance, as the grouping doesn't match our positional system's intuitive structure, where digits in the same positional groups (units, tens, hundreds) should ideally be processed together for optimal numerical reasoning and arithmetic operations.

An even more surprising development is that forcing the R2L token clustering of numbers in models already trained with L2R clustering through the use of commas in the input (ex. $12,345$) leads to big improvements in arithmetic performance (Millidge, 2024). Despite the model learning representations adapted to work with a L2R token clustering strategy, forcing a R2L clustering at inference time shows substantial improvements in arithmetic tasks, which means that despite being learned through an unfavorable tokenization approach, the numeric representations retain the properties that allow for the performance to improve when the clustering scheme is corrected.

There can be different hypotheses on why this might be, for example:

- Arithmetic operations would still work locally in the 0-999 range, which allows for a correct reading on them and possible generalization on a larger scale.
- The forced tokenization also happens in the data, as numbers are often separated by punctuation in clusters of 3 digits, right to left, for legibility reasons (Singh & Strouse, 2024)

Still, we are left with the fact that the learned representations work better for a tokenization strategy different from the one the model was trained for. At the very least, the data being biased towards a R2L representation

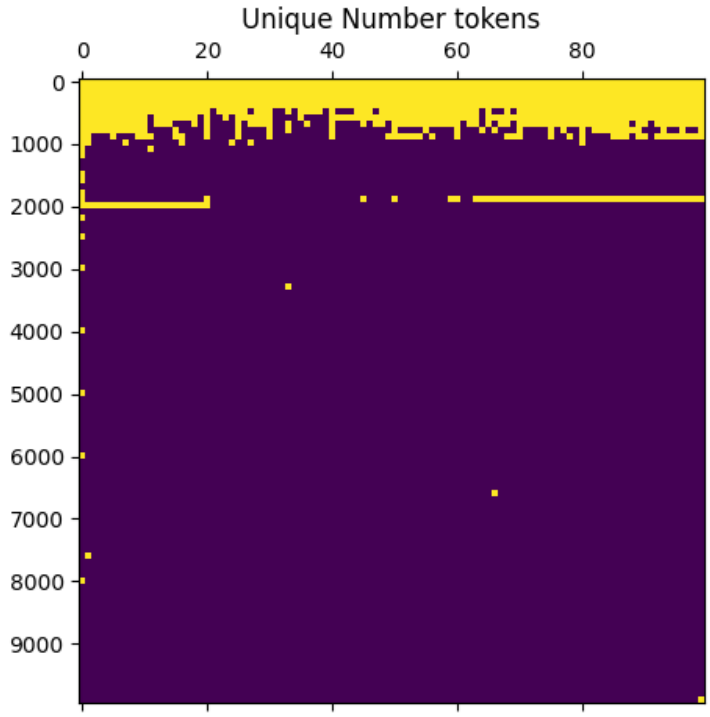


Figure 1: GPT-2 number tokenization. Each row represents 100 numbers, yellow squares mean that the number is represented by a single token, purple ones by multiple (Millidge, 2023)

(in the form of using the Arabic number system and adopting legibility rules that accommodate right to left calculations) lead to embeddings that maintain that bias even when learned in a L2R fashion. This can be a possible hint towards the optimality of certain representations compared to others, given the resilience in preferring a certain tokenization scheme over the one the model is trained on.

Table 1: Language models with their respective tokenization strategy for numbers.

Model	Strategy
LLaMA 1 & 2	single digit
LLaMA 3	L2R chunks of 3 digits
OLMo 2	L2R chunks of 3 digits
GPT-2	pure BPE
GPT-3.5/4	L2R chunks of 3 digits
Claude 3/4	R2L chunks of 3 digits

Reification as computed embeddings - xVal

There have been other, more comprehensive approaches to the improvement of the representation of numeric values. xVal is a notable one, as its approach encompasses real numbers beyond just integers and does away with learning different representation for each number.

The idea is maximizing the inductive bias in the representation by having embeddings that are computed based on the number to be represented. Numerical values represented by a single embedding vector associated with the [NUM] special token.

This fits very well with the idea of reification: the embedding is no longer just a representation, but it contains and has properties of the object it represents.

The model uses two separate heads for number and token predictions. If the token head predicts a [NUM] token as the successor, the number head gets activated and outputs a scalar. The rest of the weights in the transformer blocks are shared, allowing the learning of representations that are useful for both discrete text prediction and continuous numerical prediction. This means the model develops number-aware internal representations throughout all its layers, not just at the output. The shared weights force the model to learn features that work for both linguistic and mathematical reasoning simultaneously.

The approach is shown to improve performance over a series of other techniques, mostly using a standard notation to represent numbers. This work has been inspired by the xVal paper, with one of its initial goals being to find good representations for computed numerical embeddings.

The search for better suited representation

A case study of savant patient DT (Murray, 2010) reveals a mathematical cognitive architecture with the following characteristics:

- Has sequence-space synesthesia with a "mathematical landscape" containing numbers 0-9999
- Each number possesses specific colors, textures, sizes, and sometimes movements or sounds
- Prime numbers have distinctive object properties that distinguish them from other numbers
- Arithmetic calculations happen automatically - solutions appear as part of his visual landscape without conscious effort
- fMRI studies showed that even unstructured number sequences had coherent visual structure for DT

Sequence-space synesthesia involves the spontaneous visualization of numerical sequences in organized spatial arrangements. The remarkable mathematical abilities of savants with this condition suggest that their specialized perceptual representations confer significant computational advantages over conventional symbolic processing.

Mottron et al. (Mottron et al., 2006) hypothesize that savant capabilities emerge from privileged access to lower-level perceptual processing systems that have been functionally re-dedicated to symbolic material processing. Under this framework, mathematical savants may bypass high-level algorithmic reasoning entirely, instead leveraging perceptual mechanisms that directly recognize patterns in numerical relationships—much like how we instantly recognize a face without consciously processing its individual features.

This raises a compelling question for artificial intelligence: Do large language models spontaneously develop similar specialized mathematical representations during training? If so, understanding these representations could inform approaches for enhancing mathematical reasoning capabilities in AI systems.

We investigate whether state-of-the-art language models develop systematic mathematical property detectors within their embedding spaces, analogous to the specialized numerical representations observed in savant cognition. Specifically, we examine whether numerical tokens are organized according to mathematical properties such as magnitude, primality, digit structure, and sequence patterns like the Fibonacci series.

Embeddings Analysis

The analytic part of this work consists in the search for structure in LLM numerical embeddings.

As stated previously, recent open source models mostly employ an L2R tokenization scheme. There are no large scale open source models using R2L tokenization as of the time of writing, but the improvement in performance observed when using R2L tokenization in L2R-trained models could be a hint that L2R embedding representations still have similar properties to the R2L ones.

We're looking for clues of mathematical properties being encoded in the embeddings. As the results show, we find strong correlations between embedding dimensions and mathematical properties, the strongest one being magnitude (data) and a very interesting one being the distance between the number and their closest Fibonacci number, with the catch that further study is needed to see if this strong connection comes from confounding factors.

Methodology

For each model, we extracted the embeddings corresponding to integer numbers representable by a single token. The embeddings were then analyzed using dimensionality reduction techniques (PCA, SVD, t-SNE and UMAP) to garner statistics and produce visualizations that could potentially highlight structures in the data. As a final stage, the embeddings were directly tested for correlations with mathematical properties of the number:

- magnitude
- relation to digit count
- primality
- evenness
- being a perfect square
- being a Fibonacci number

For binary properties, the correlation was measured with vectors with value 1 for the indexes corresponding to numbers where the property is true and 0 elsewhere. To account for the continuous nature of the embeddings, smoother functions that might relate to the same properties were also taken into account:

- "squareness": an approximate measure of closeness of the number to a perfect square

$$\text{squareness}(n) = \begin{cases} 0 & \text{if } n \leq 0 \\ 1 - 2 \cdot \min\left(\sqrt{n} - \lfloor\sqrt{n}\rfloor,\ \lceil\sqrt{n}\rceil - \sqrt{n}\right) & \text{if } n > 0 \end{cases}$$

- prime proximity: distance between the number and the nearest perfect prime, measured in integers between the two.
- Fibonacci proximity: distance between the number and the nearest Fibonacci number, measured in integers between the two.

We took two models in consideration:

- OLMo-2-1124-7B is a model by AllenAI, which is favorable to research uses thanks to the full disclosure of training data, code, logs and checkpoints
- Llama-3.2-1B-Instruct, due to being a small and manageable model to do analysis with on limited hardware

OLMo-2-1124-7B

Linear analysis

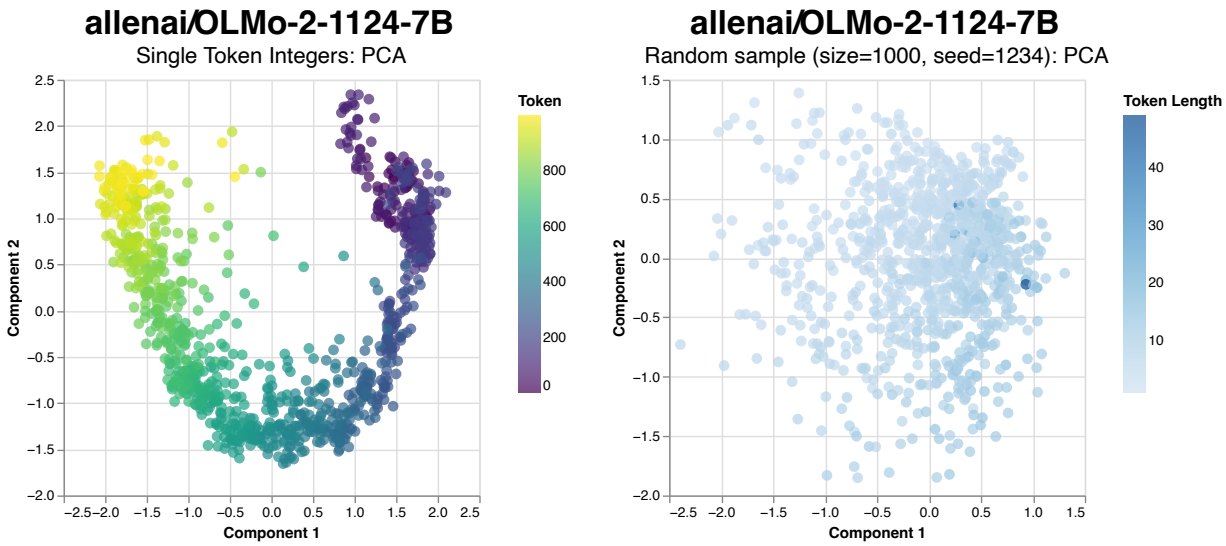


Figure 2: Principal components 1 and 2 of the OLMo model. Random embeddings sample for comparison.

Projecting the numerical token embeddings onto the first two principal components reveals a U-shaped curve. This structure constitutes a one-dimensional manifold embedded within the two-dimensional principal component space.

The manifold structure is particularly significant because it demonstrates that numerical tokens do not occupy the embedding space randomly. Instead, they follow a constrained path that preserves numerical relationships, suggesting that the model has learned to encode ordering properties of the numbers within its representation. The gradient is particularly smooth, suggesting that similar numbers maintain spatial proximity in the reduced space.

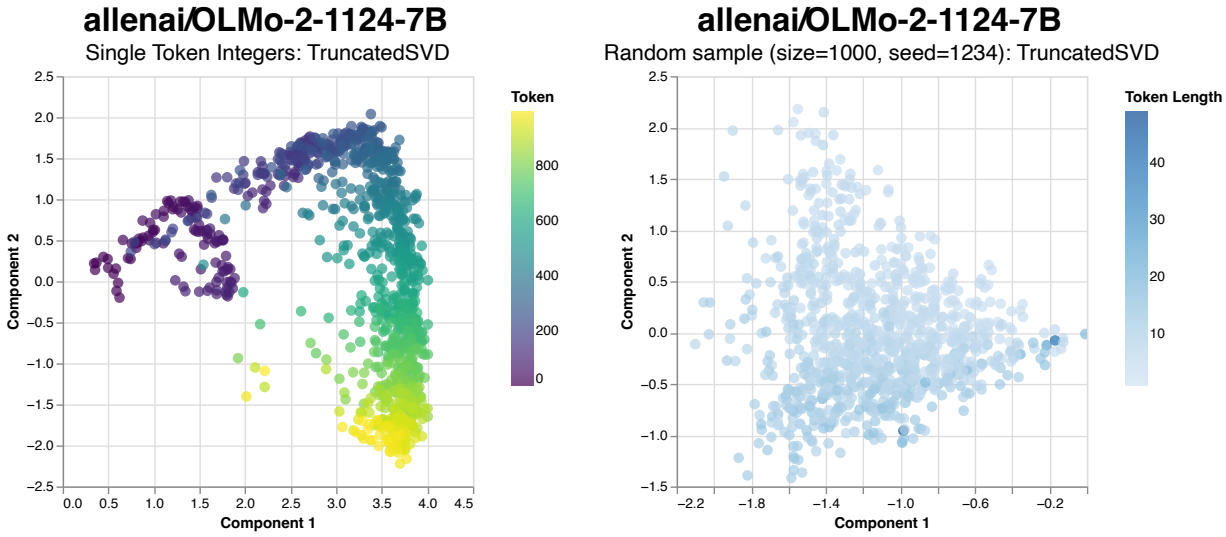


Figure 3: SVD for the two main components of the OLMo model, with random embeddings sample for comparison

The SVD visualization, lacking the data centering done in the PCA, shows a much more consistent geometric structure, suggesting that the encoding of information might be done in absolute distances rather than just with relative positioning between data points. There is also a very notable recursive, fractal structure, repeating itself for numbers with one, two and three digits.

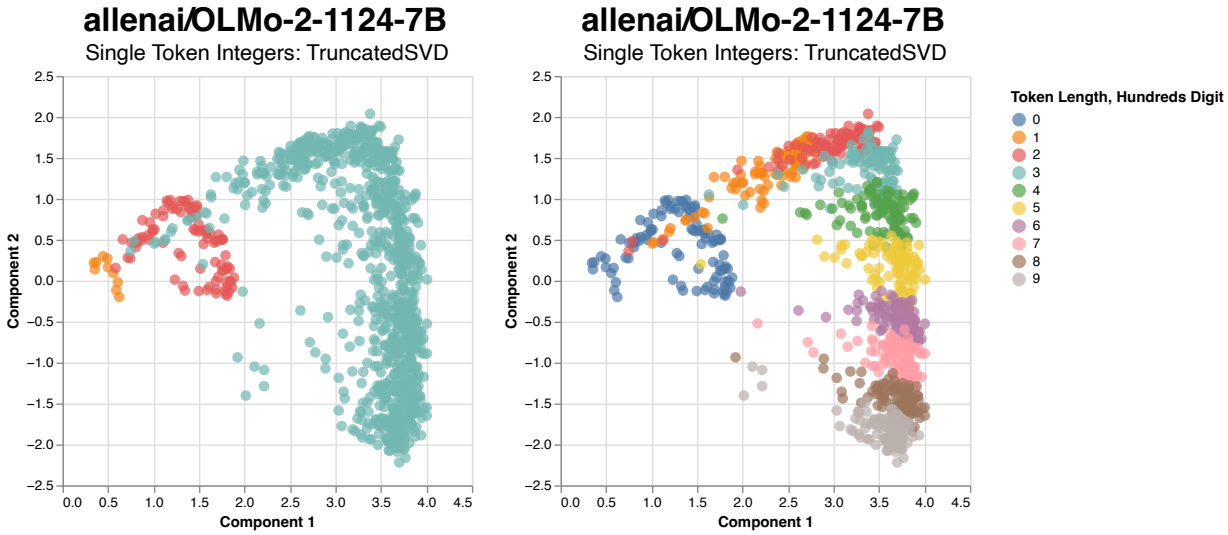


Figure 4: SVD coloring done by digit length and hundreds digit, highlighting the clustering properties of the embeddings.

Explained variance

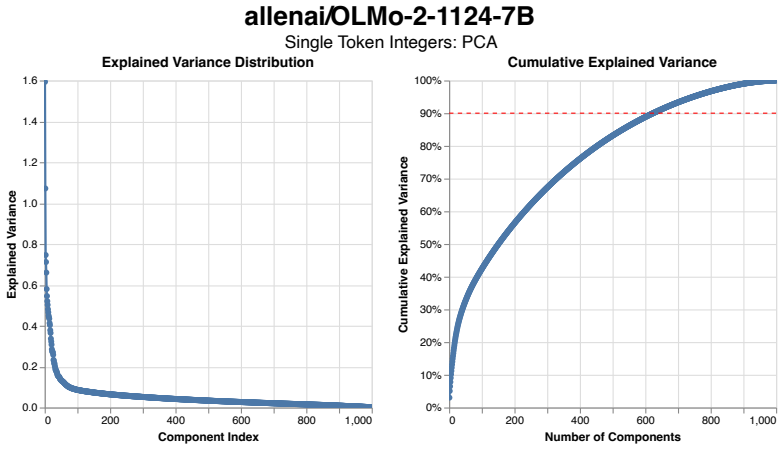


Figure 5: OLMo PCA - explained variance overview

The explained variance by component plot (left) shows a sharp drop within the first few components, meaning that the first principal components capture dramatically more variance than subsequent ones. The cumulative explained variance shows that approximatively 600 principal components are needed to reach 90% of explained variance.

By this we can conclude that the embeddings have a much lower intrinsic dimensionality than their full 4096 dimensions, and that they lie on a low-dimensional manifold in the full representation space. Only one-fifth of the total embedding space is necessary to capture 90% of the variance.

Non-linear analysis

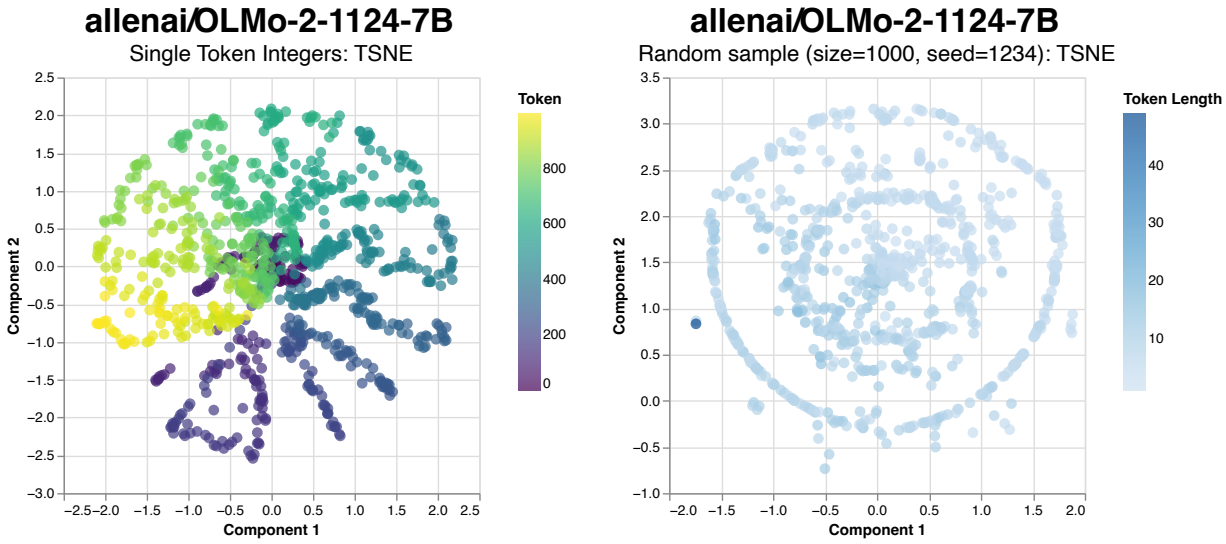


Figure 6: t-SNE visualization for OLMo embeddings.

Table 2: t-SNE hyperparameters for the presented plots.

Parameter	Value
perplexity	75
max_iter	3000
learning_rate	50
early_exaggeration	20
random_state	42

The t-SNE visualization shows a distinctive branching pattern emanating from a central region, with low numbers at the center and higher ones radiating outward. The color progression follows these branches, indicating that numerical sequences are preserved along each arm. The gradient seems also to transition circularly; branches with gradually increasing numbers turn around the center before abruptly getting back to the start. When interpreting the colors as indicators of depth, it can look like a spiral from a top-down perspective.

UMAP has been run using both Euclidean and cosine distances, since the SVD visualization has shown that absolute distances can matter in this model. In the UMAP case we can observe a loss of shape similar to what happened in the PCA and SVD case. While the structure is congruent when using Euclidean distances, segregated clusters form when representing cosine similarity, with their predominant criterion of division being the hundreds' digit. Using Euclidean distances gives a picture similar to t-SNE, but projected and stretched and with more dispersion for numbers close to zero. The spiral-like conformation is also notable here.

Correlation with mathematical properties

By taking all the components and their correlations with the properties we're testing for, we're able to find the most correlative component-property pairs. Most of the components that exhibit a strong correlation does so in terms of their magnitude (measured as the correlation with the log_{10} of the number considered).

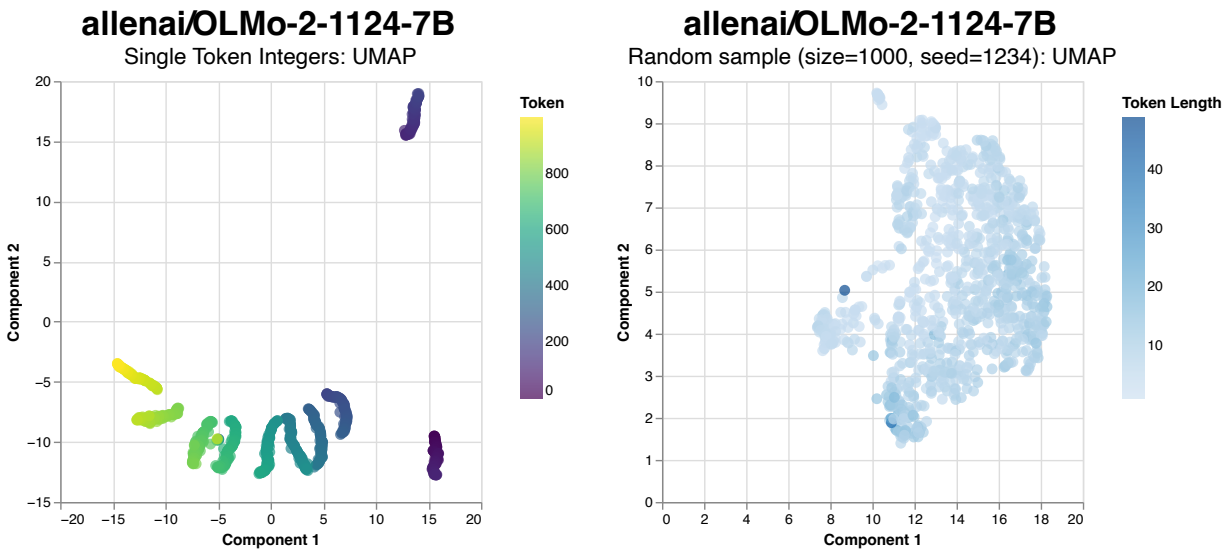


Figure 7: UMAP visualization with cosine distance

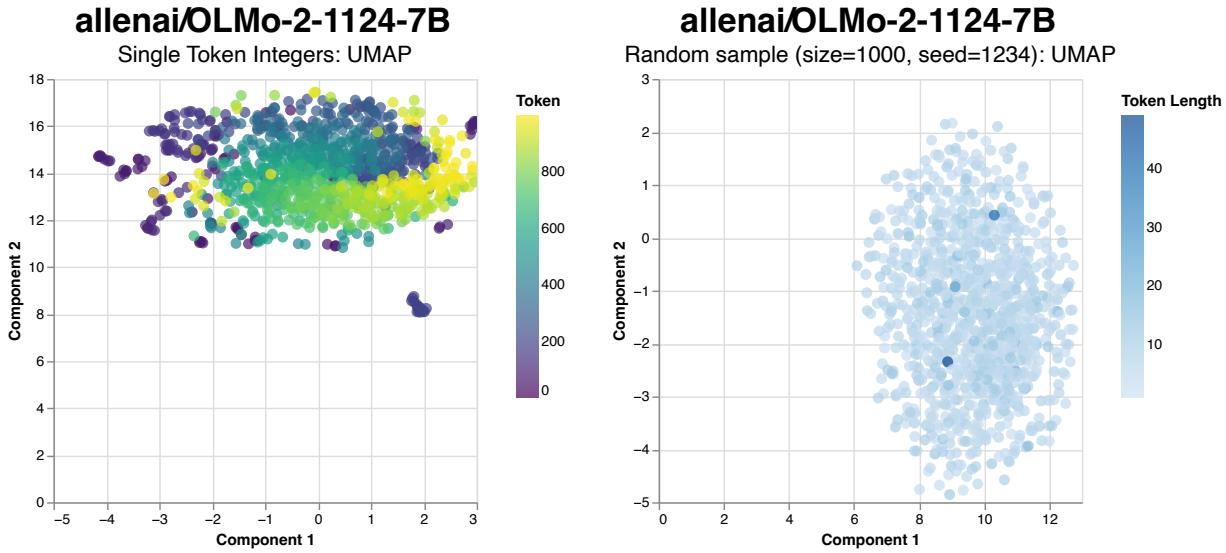


Figure 8: UMAP visualization with Euclidean distance

Table 3: The most correlated component-property pairs.

Dimension	Property	Correlation	P_Value
514	magnitude	-0.672870	8.446593e-133
3085	magnitude	-0.606534	1.646729e-101
2538	magnitude	-0.567317	3.016150e-86
665	magnitude	-0.500278	1.891640e-64
514	digit_count	-0.484543	5.301794e-60
1012	magnitude	-0.475346	1.653308e-57
1012	digit_count	-0.462706	3.346741e-54
2538	digit_count	-0.454433	4.115639e-52
3085	digit_count	-0.451884	1.766273e-51
3879	magnitude	0.446828	3.059244e-50
110	magnitude	0.445026	8.359542e-50
1820	magnitude	0.430705	1.991192e-46
1107	magnitude	-0.428056	8.055688e-46
3502	magnitude	0.425927	2.456033e-45
421	magnitude	-0.423685	7.871414e-45
90	magnitude	0.419583	6.487704e-44
3548	magnitude	0.411402	3.991055e-42
1554	magnitude	-0.409980	8.075082e-42
3085	fibonacci_proximity	0.409860	8.566550e-42

Magnitude and digit count would be expected to be widely encoded, and they seem in fact the dominant factor (also, they would be correlated with each other). The most interesting property shown here is definitely Fibonacci_proximity, representing the distance between the number and the closest Fibonacci number. Having a correlation index of 0.409 with a very small p-value would be a strong indicator that this is an important factor in the encoding of the embeddings. However, after further consideration it was noticed that can be explained by the strong correlation between the Fibonacci proximity and magnitude itself (≈ 0.547, p-value $< 1e-79$). This confounding factor might make the correlation by itself inconclusive, and further research would be needed to establish the connection between the two quantities. There are also two strong correlation with both the is_fibonacci and the is_prime property, which shows the embeddings are likely encoding some information about the primality of the number considered and their relationship to the Fibonacci series.

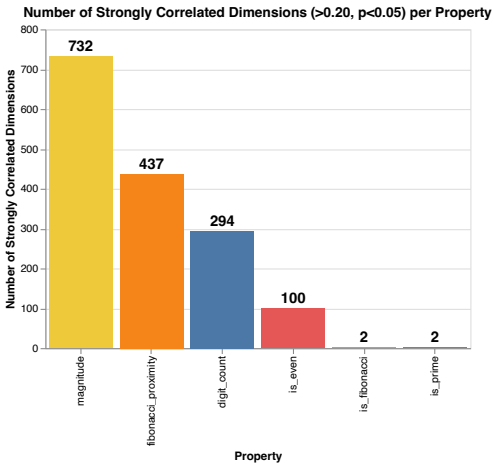


Figure 9: Mathematical properties with the number of associated strongly correlated dimensions

Llama-3.2-1B-Instruct

Linear analysis

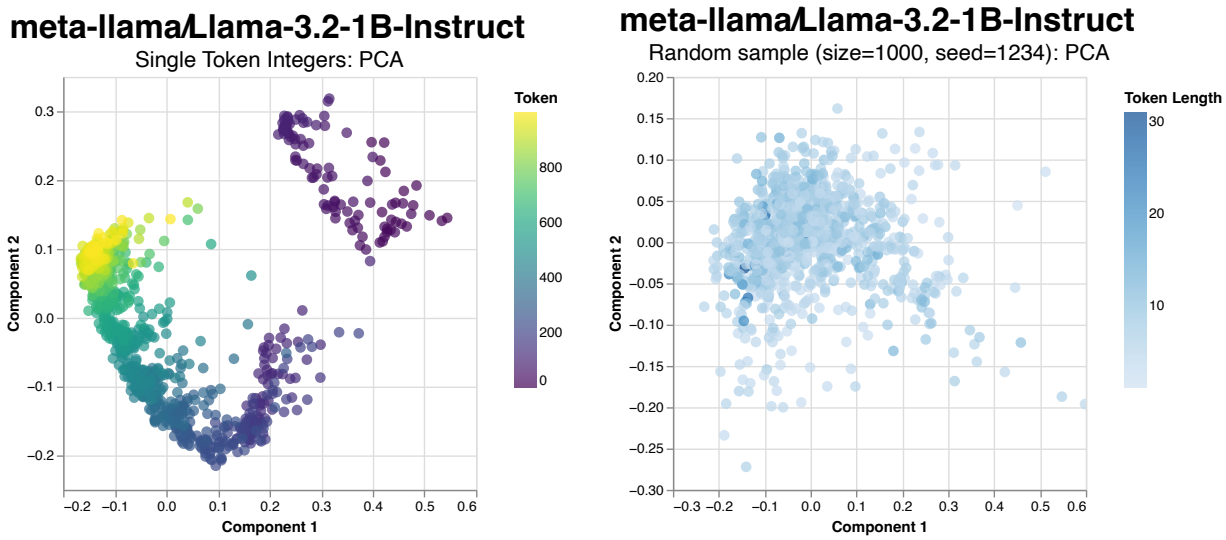


Figure 10: PCA visualization of Llama embeddings.

The PCA projection shows a continuous, arc-shaped curved manifold, with smoother transitions between numbers and a distinct separation with numbers close to 0. As with what was seen with OLMo, it looks like the PCA centering might end up destroying geometric relationships that are better preserved in the SVD visualizations.

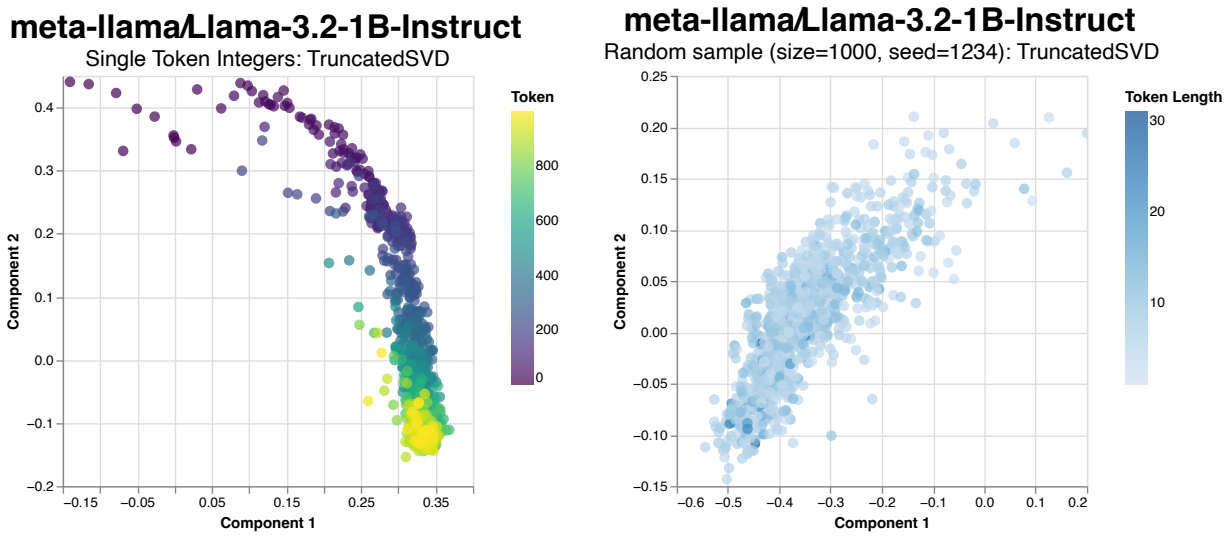


Figure 11: SVD visualization of Llama embeddings

The SVD plot shows a remarkably linear arrangement - numbers form an almost straight diagonal line from small (yellow) to large (purple) values. This linear structure is much more pronounced than OLMo-2's curved SVD patterns, and it is a unique shape rather than a recursive, recurring pattern.

The digit-based coloring reveals clear but subtle clustering by mathematical properties. Unlike OLMo-2's

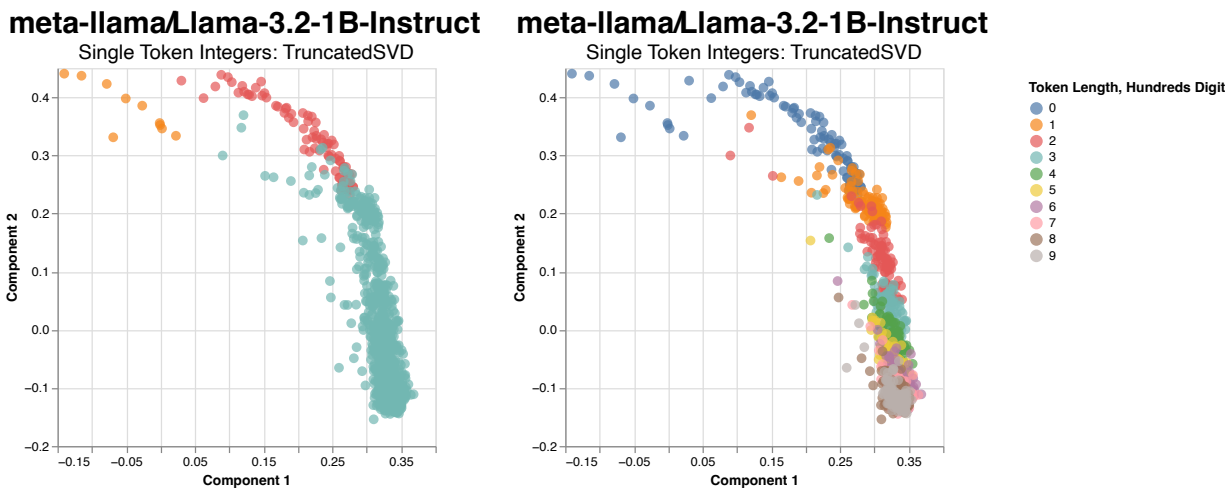


Figure 12: Llama SVD visualization by digit

distinct spatial territories, Llama-3.2 shows gradual transitions along the linear arrangement while maintaining digit-based organization patterns.

Explained variance

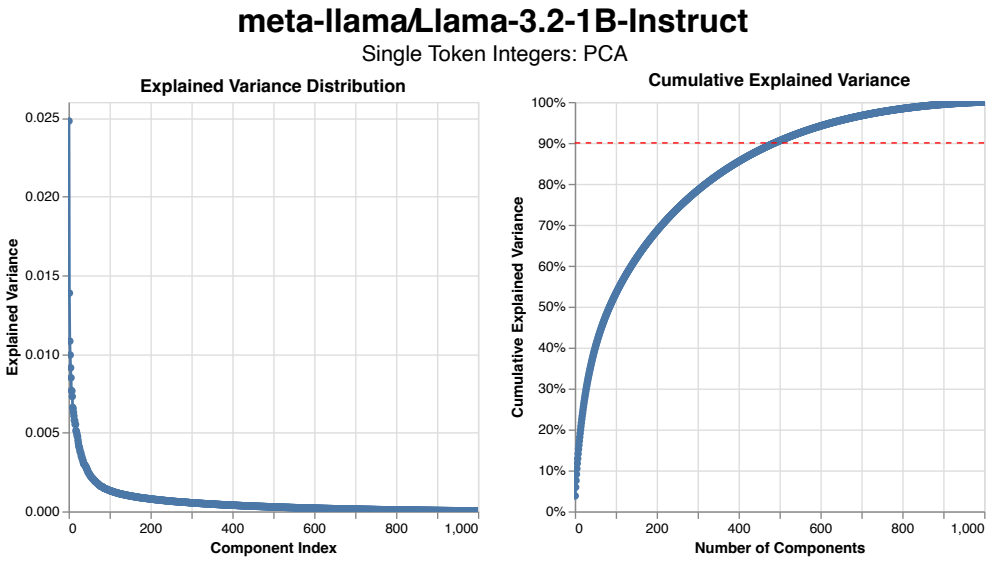


Figure 13: Llama PCA explained variance.

The explained variance plot reveals slightly higher information concentration than OLMo-2. Llama-3.2 reaches 90% explained variance with approximately 500 components compared to OLMo-2's 500 components. This suggests more efficient numerical encoding in the smaller model.

Non-Linear analysis

These nonlinear projections reveal dramatically different organizational patterns from both the linear methods and from OLMo-2's structures.

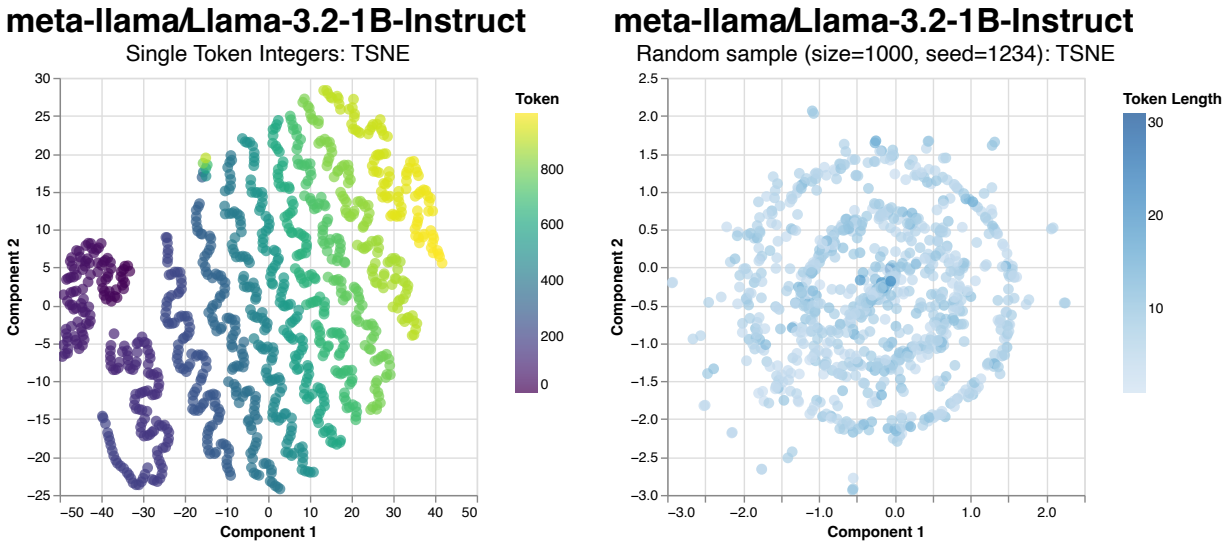


Figure 14: t-SNE structure in Llama

The t-SNE visualization is very unusual, and show continuous, winding structures that might look like they had been uncoiled or unwound from a higher-dimensional spiral arrangement. The mathematical progression follows these winding paths smoothly. This can be informative, as for their particularly keen encoding of the Fibonacci sequence, as will be shown successively.

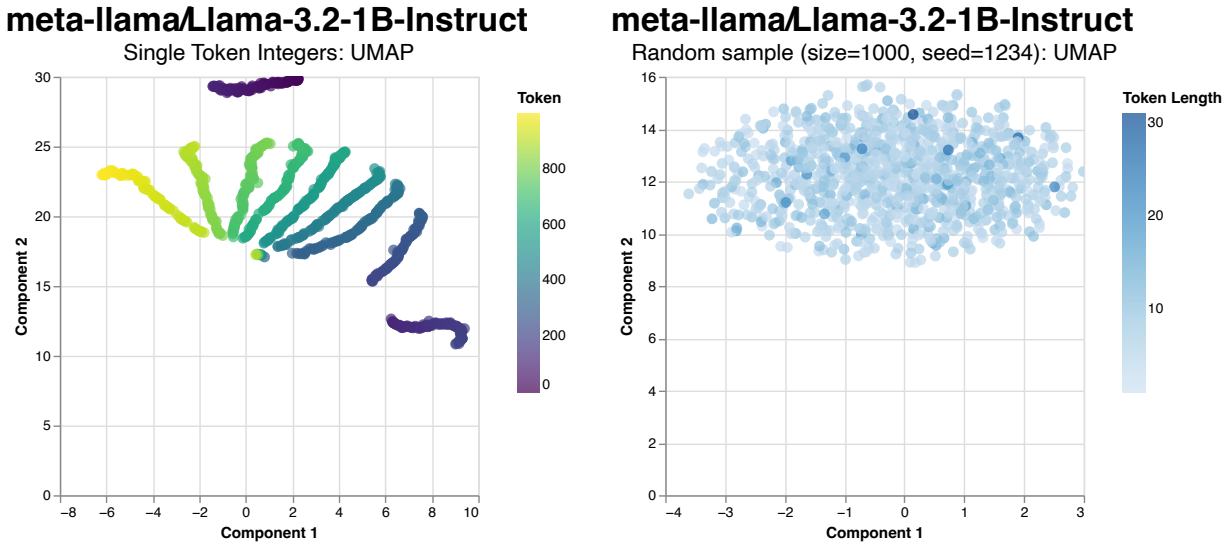
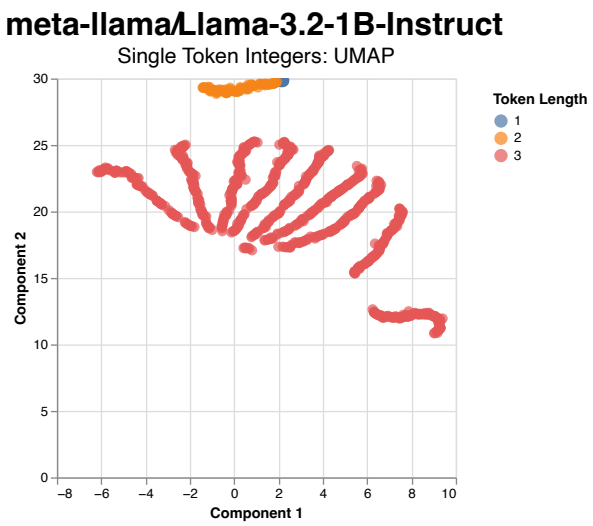
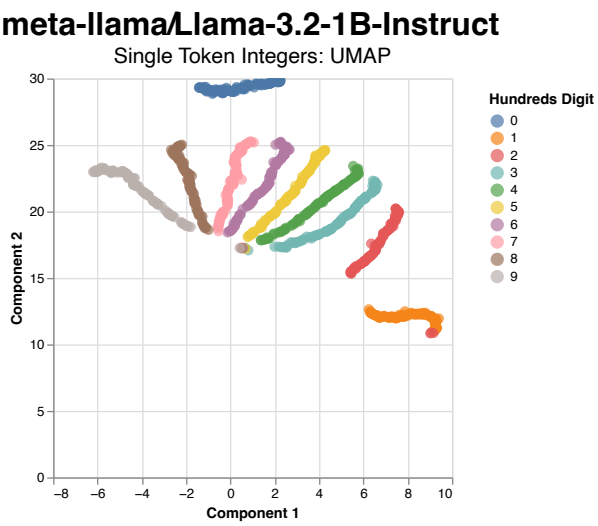
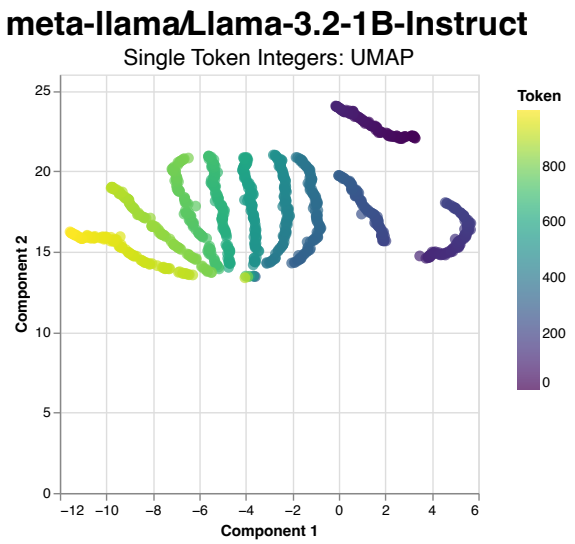
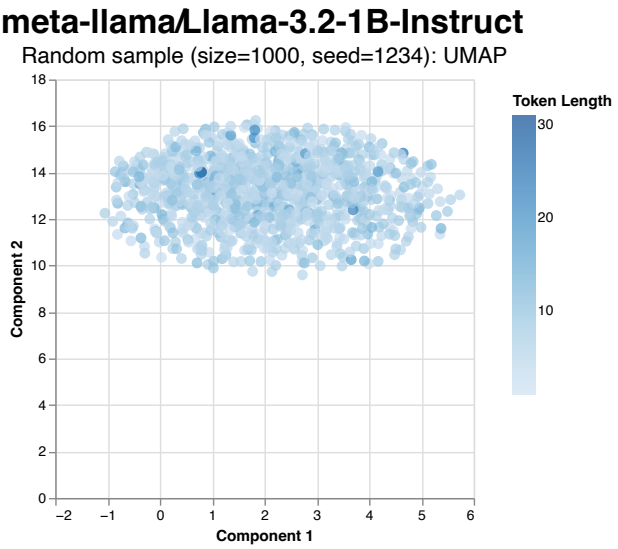


Figure 15: UMAP with cosine similarity in Llama









The UMAP visualization is resembling the OLMo's one. It's also interesting to see that changing the distance function to Euclidean doesn't have particular effects, unlike the previous OLMo visualization.

Correlation with mathematical properties

In this case we see a lot more components directly encoding for digit_count, as well as for parity. There are 12 strongly correlated components with primality and 10 with being a Fibonacci number. There is still a big number of components strongly correlated with the fibonacci_proximity, which would need further analysis to fully establish whether their sensitivity to magnitude dominates over the detection of Fibonacci numbers.

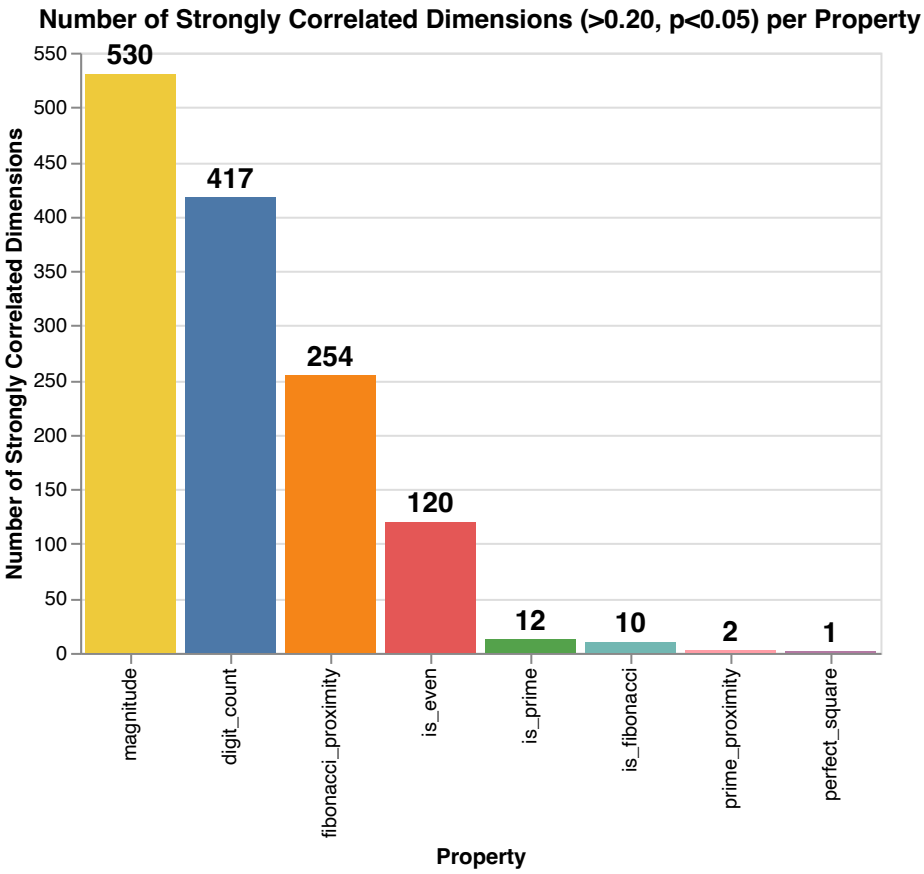


Figure 16: Dimensions strongly correlated with properties in Llama 3.2

Conclusions

This thesis investigated whether Large Language Models develop structured numerical representations that might share organizational principles with the specialized representations observed in mathematical savants. Through analysis of numerical embeddings in OLMo-2-1124-7B and Llama-3.2-1B-Instruct, we found evidence of systematic mathematical structure within learned representations.

Key Findings

Structured Numerical Embeddings

Our analysis revealed that numerical tokens are not randomly distributed in embedding space but follow organized patterns:

Geometric Organization: Principal component analysis showed that numerical embeddings lie on low-dimensional manifolds. OLMo-2 exhibited U-shaped curves with recursive patterns across digit ranges, while Llama-3.2 displayed more linear arrangements. Both models required only ~500 components to capture 90% of variance, indicating substantial dimensionality reduction from the full 4096-dimensional space.

Mathematical Property Correlations: Multiple embedding dimensions showed significant correlations with mathematical properties: - Magnitude and digit count exhibited the strongest correlations ($r > 0.67$) - Primality was encoded across multiple dimensions - Fibonacci number proximity showed notable correlations, though potentially influenced by magnitude effects - Binary properties like evenness were systematically represented

The consistent organization of numerical embeddings according to mathematical properties suggests that neural language models might spontaneously develop structured representations during training. This organization goes beyond simple ordering, incorporating complex mathematical relationships like primality and sequence membership.

Limitations

The analysis was limited to two models and a specific set of mathematical properties. The relationship between Fibonacci proximity and magnitude highlights the challenge of isolating specific property detectors from general ordering mechanisms. Further investigation with controlled experiments would be needed to establish causal relationships.

Whether these findings extend to larger models, different architectures, or other mathematical domains remains to be determined. The observed structures may reflect training data properties as much as emergent organizational principles.

Future Directions

This work suggests several research directions: investigating mathematical representations across model scales and architectures, exploring computed embedding approaches that leverage discovered geometric structures, and examining whether similar organizational principles apply to other domains where specialized representations might be beneficial.

The findings contribute to understanding how neural language models organize mathematical information and suggest that structured representations may emerge naturally in systems trained on numerical data. While modest in scope, these results provide a foundation for further investigation into the relationship between representational structure and mathematical reasoning capabilities in artificial systems.

Bibliography

Ali, M., Fromm, M., Thellmann, K., Rutmann, R., Lübbering, M., Leveling, J., Klug, K., Ebert, J., Doll, N., Buschhoff, J. S., Jain, C., Weber, A. A., Jurkschat, L., Abdelwahab, H., John, C., Suarez, P. O., Ostendorff, M., Weinbach, S., Sifa, R., … Flores-Herr, N. (2024, March 17). *Tokenizer Choice For LLM Training: Negligible or Crucial?* https://doi.org/10.48550/arXiv.2310.08754

Cowan, R., & Frith, C. (2009). Do calendrical savants use calculation to answer date questions? A functional magnetic resonance imaging study. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *364*(1522), 1417–1424. https://doi.org/10.1098/rstb.2008.0323

Millidge, B. (2023, February 4). *Integer tokenization is insane*. http://www.beren.io/2023-02-04-Integer-tokenization-is-insane/

Millidge, B. (2024, July 7). *Right to Left (R2L) Integer Tokenization*. http://www.beren.io/2024-07-07-Right-to-Left-Integer-Tokenization/

Mottron, L., Lemmens, K., Gagnon, L., & Seron, X. (2006). Non-algorithmic access to calendar information in a calendar calculator with autism. *Journal of Autism and Developmental Disorders*, *36*(2), 239–247. https://doi.org/10.1007/s10803-005-0059-9

Murray, A. L. (2010). Can the existence of highly accessible concrete representations explain savant skills? Some insights from synaesthesia. *Medical Hypotheses*, *74*(6), 1006–1012. https://doi.org/10.1016/j.mehy.2010.01.014

Singh, A. K., & Strouse, D. J. (2024, February 22). *Tokenization counts: The impact of tokenization on arithmetic in frontier LLMs*. https://doi.org/10.48550/arXiv.2402.14903

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023, August 1). *Attention Is All You Need*. http://arxiv.org/abs/1706.03762