# Monotonic Representation of Numeric Properties in Language Models

**Benjamin Heinzerling**[1, 2] and **Kentaro Inui**[3, 2, 1]
[1]RIKEN [2]Tohoku University [3]MBZUAI
benjamin.heinzerling@riken.jp | kentaro.inui@mbzuai.ac.ae

## Abstract

Language models (LMs) can express factual knowledge involving numeric properties such as *Karl Popper was born in 1902*. However, how this information is encoded in the model's internal representations is not understood well. Here, we introduce a method for finding and editing representations of numeric properties such as an entity's birth year. We find directions that encode numeric properties monotonically, in an interpretable fashion. When editing representations along these directions, LM output changes accordingly. For example, by patching activations along a "birthyear" direction we can make the LM express an increasingly late birthyear. Property-encoding directions exist across several numeric properties in all models under consideration, suggesting the possibility that monotonic representation of numeric properties consistently emerges during LM pretraining. Code: https://github.com/bheinzerling/numeric-property-repr

A long version of this short paper is available at: https://arxiv.org/abs/2403.10381

## 1 Introduction

Language models (LMs) can express factual knowledge (Petroni et al., 2019; Jiang et al., 2020; Roberts et al., 2020; Heinzerling and Inui, 2021; Kassner et al., 2021). For example, when queried *In which year was Karl Popper born?* Llama 2 (Touvron et al., 2023) gives the correct answer *1902*. While the question if LMs "know" anything at all is subject of debate (Bender and Koller, 2020; Hase et al., 2023b; Mollo and Millière, 2023; Lederman and Mahowald, 2024), empirical work has progressed from behavioral analysis focused on the accuracy and robustness of knowledge expression (Shin et al., 2020; Jiang et al., 2021; Zhong et al., 2021; Youssef et al., 2023) to representational analysis aimed at understanding how factual knowledge is encoded[1] in model parameters (De Cao et al., 2021; Mitchell et al., 2021; Meng et al., 2022) and activations (Hernandez et al., 2023; Merullo et al., 2023; Geva et al., 2023; Gurnee and Tegmark, 2023).

However, representational analysis has mainly targeted entity-entity relations such as *Warsaw is the capital of Poland*. How LMs encode factual knowledge involving numeric properties, such as an entity's birthyear, is less understood. Unlike entity-entity relations, numeric properties have natural ordering and monotonic structure. While this structure is intuitive for humans, LMs encounter numeric properties mostly in form of unstructured textual mentions. This raises the question if LMs learn to represent numeric properties appropriately, according to their structure.

Here, we devise a simple method for identifying and manipulating representations of numeric properties in LMs. We find low-dimensional subspaces that strongly correlate with numeric properties across models and numeric properties, thereby confirming and extending prior observations of representations of numeric properties in LMs (Liétard et al., 2021; Faisal and Anastasopoulos, 2023; Gurnee and Tegmark, 2023; Godey et al., 2024). Going beyond prior work (see §A), we show that by causally intervening along directions in these subspaces, LM output changes correspondingly. That is, we find a monotonic relationship between the intervention and the quantity expressed by the LM. For example, an entity's year of birth shifts according to the strength and sign of the intervention along a "birthyear" direction (Fig. 1). Taken together, our findings suggest that LMs represent numeric properties in a way that reflects their natural structure and that such monotonic representations consistently emerge during LM pretraining.

[1]We say "X is encoded in Y" as shorthand for "X can be easily extracted from Y". See caveats in §5.



*Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 175–195
August 11-16, 2024 

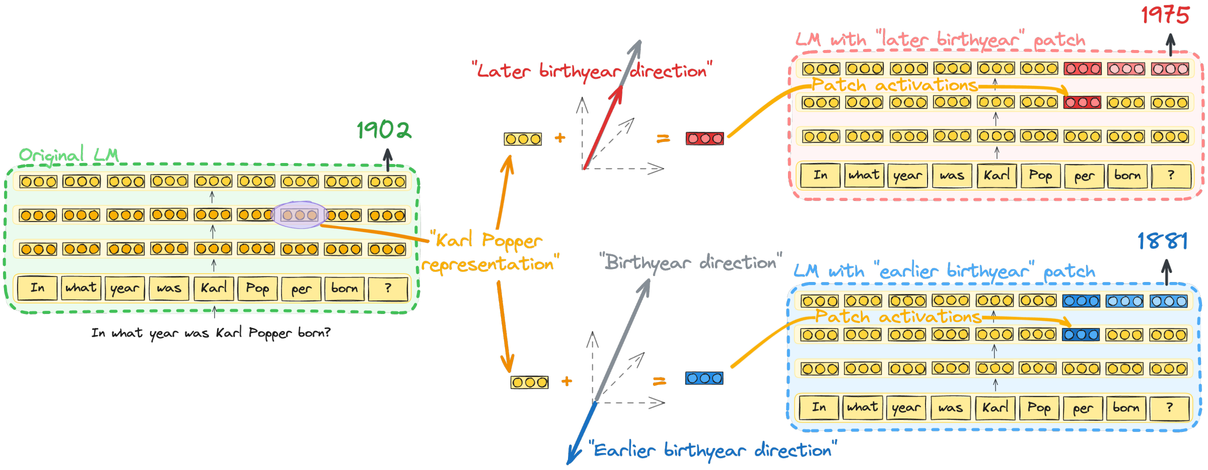



Figure 1: Sketch of our main finding. Patching entity representations along specific directions in activation space yields corresponding changes in model output.

**Terminology.** We briefly clarify important terms. A **quantity** consists of a scalar numeric **value** paired with a **unit** of measurement. A **numeric property** is a property that can be described by a quantity, e.g., birthyear, population size, geographic latitude. A **numeric attribute** is an instance of a numeric property, associated with a particular entity. For example, Karl Popper has the numeric attribute `birthyear:1902`. By **linear representation** we denote the idea that a numeric attribute is encoded in a linear subspace of a LM's activation space. A **monotonic representation** is a linear representation characterized by a monotonic relationship between directions in activation space and the value of the encoded numeric attribute. That is, as activations shift along a particular direction the value of the corresponding numeric attribute increases or decreases monotonically.

## 2 Finding Property-Encoding Directions

**Motivation.** While numeric properties generally map naturally onto simple canonical structures, such as number lines or coordinate systems, it is not obvious that pretraining on largely unstructured data enables LMs to appropriately represent such structures. Our goal is to find out if and how numeric properties are encoded in the geometry of LM representations. How could such an encoding look like? Based on two arguments, we hypothesize that numeric properties are encoded in low-dimensional linear subspaces of activation space.

The first argument rests on a key principle in representation learning: a model generalizes if and only if its representations reflect the structure of the data (Conant and Ashby, 1970; Liu et al., 2022). To the degree that current LMs generalize, in the sense of achieving non-trivial performance on benchmarks involving knowledge of numeric properties (Petroni et al., 2019), we can expect that their representations reflect the structure of numeric properties. Since the natural structure of many numeric properties is low-dimensional, we expect to find low-dimensional structure in the representations of a well-performing model.

As second argument we adduce the linear representation hypothesis, which posits a correspondence between concepts and linear subspaces (Elhage et al., 2022; Park et al., 2023; Nanda et al., 2023). If the linear representation hypothesis is true,[2] this would imply that numeric properties are encoded in linear subspaces. For brevity, we will call a low-dimensional linear subspace of a LM's activation space a *direction*, regardless of whether it is one- or multi-dimensional.

**Method.** Motivated by the hypothesis that numeric properties are encoded as directions in activation space, we now devise an experimental setup for finding out if such directions exist. A common choice for identifying linear structure is principal component analysis (PCA; Pearson, 1901). However, PCA looks for directions of maximum variance, while we want to find directions that maximally covary with model outputs. This kind of problem can be solved with partial least squares regression (PLS; Wold et al., 2001).

Concretely, for a given numeric property we collect $n$ entities that have this property. For each

[2]For positive evidence, see Marks and Tegmark (2023); Merullo et al. (2023); Tigges et al. (2023); Jiang et al. (2024)

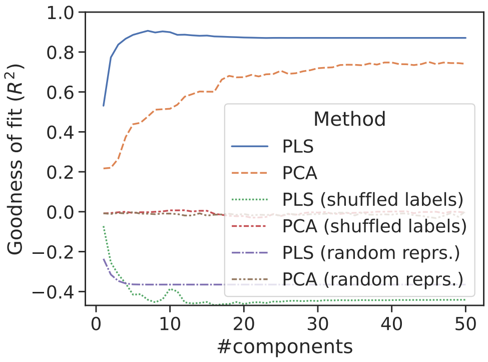



Figure 2: Low-dimensional subspaces of Llama-2-13B's activation space are predictive of the quantity expressed by the LM when queried for an entity's birthyear. Each line shows the performance of a regression model fitted to predict the expressed birthyear from LM representations, as a function of the number of PCA/PLS components. Unlike PCA regression (dashed orange), PLS (solid blue) identifies a small set of predictive components. Controls with shuffled labels and random representations fail to find predictive subspaces.

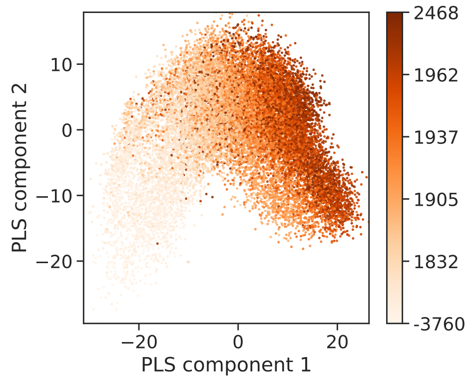



Figure 3: Projection onto the top two PLS components reveals monotonic structure in LM representations. Dots represent entities and color corresponding birthyears.

entity $e$ we encode a prompt with a LM to obtain entity representation $x_e$ of dimension $d$. That is, $X = [x_1 \cdots x_n]^\intercal \in \mathbb{R}^{n \times d}$. We also collect the quantity $y_e$ expressed by the LM, i.e., $Y = [y_1 \cdots y_n]^\intercal \in \mathbb{R}^n$. Having collected entity representations $X$ and associated LM outputs $Y$, we fit a $k$-component PLS model to predict $Y$ from a $k$-dimensional subspace of $X$. We vary the number of components $k$ and record goodness of fit via the coefficient of determination $R^2$.

**Results.** After selecting six frequent numeric properties in Wikidata (Vrandečić and Krötzsch, 2014), for each property we sample $n = 1000$ popular[3] entities and prompt the LM (in English) for the corresponding attribute (See samples of entities and prompts in §B). As entity representation we take the hidden state of the entity mention's last token at layer $l$, choosing $l$ as described in §F.

PLS regression results for Llama 2 13B representations are shown in Fig. 2 and results for additional models in §C. All properties can be predicted well ($R^2 \geq 0.79$), with the exception of elevation ($R^2 = 0.43$). Across all six properties, PLS identifies small sets of predictive components. For example, a PLS model with $k = 7$ components achieves a goodness of fit of $R^2 = 0.91$ when predicting birthyear attributes from entity representations. Generally, all LMs appear to encode almost the entirety (95% of maximum $R^2$) of their stored numeric attribute information in two- to six-dimensional subspaces (see §D).

To further illustrate the low dimensionality of numeric property representation, we plot a projection onto the top two PLS components for the birthyear property in Fig. 3 and for more properties and models in §E. Most plots show directions along which attribute values increase monotonically, reflecting good PLS fit for the corresponding properties.

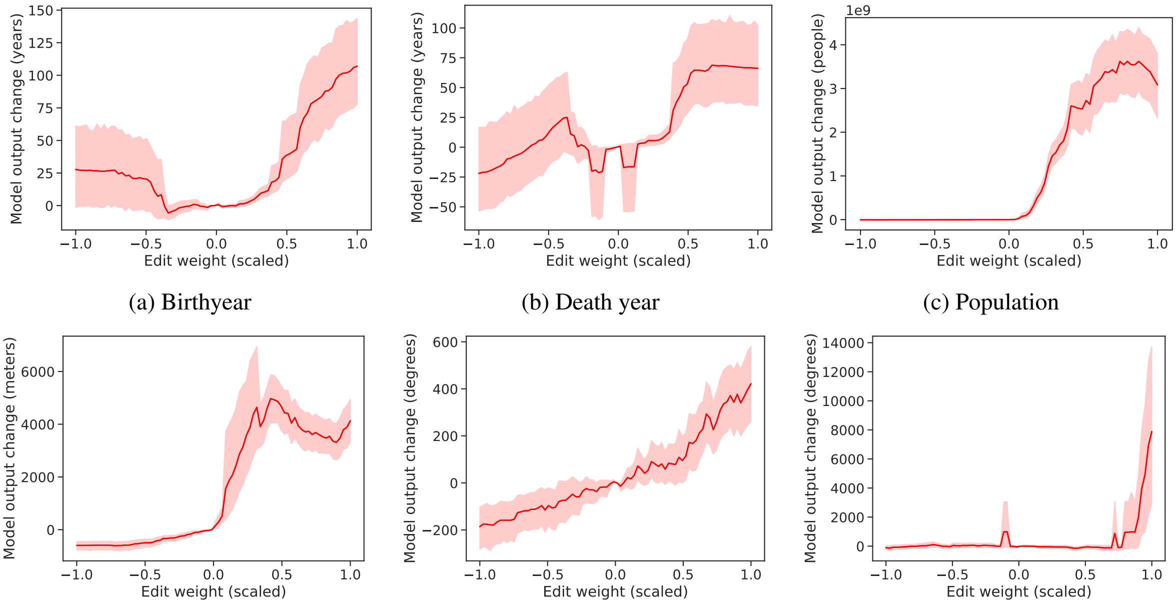
## 3 Effect of Property-Encoding Directions

**Motivation.** So far, we have found correlative evidence for the existence of directions in activation space that monotonically encode numeric properties. However, representation is not a sufficient criterion for computation (Lasri et al., 2022). In our case this means that numeric properties might be encoded in representations without affecting model output. In order to make the stronger claim that numeric properties are not only encoded monotonically, but that these representations have a monotonic effect on LM output, we now perform interventions to establish causality.

Intuitively, we want to find out if making "small" interventions leads to small changes in model output, if "large" interventions lead to large changes, and if the sign of the intervention matches the sign of the change. We now formalize this intuition by adapting the definition of linear representation by Park et al. (2023) and Jiang et al. (2024).

**Definition 1** (Linear representation of numeric properties, adapted from Jiang et al. (2024))**.** A numeric property is represented linearly if for all pairs

[3]We define popular entities as those in the top decile of the rank mean of Wikidata degree and Wikipedia article length.




(a) Birthyear (b) Death year (c) Population

(d) Elevation (e) Latitude (f) Longitude

Figure 4: Effect of activation patching along property-specific directions across six numeric properties. Each subplot shows the change in the numeric attribute value expressed by Llama 2 13B, as a function of the edit weight $\alpha_s$. Red lines show means across 100 entities and bands indicate standard deviations.

of attribute instances $i, j$ with quantities $q_i \neq q_j$ and their representations $\vec{x_i}, \vec{x_j}$, there exists a *steering vector* $\vec{u}$ so that $\vec{x_i} - \vec{x_j} \in \text{Cone}(\vec{u})$, where $\text{Cone}(\vec{v}) = \{\alpha\vec{v} : \alpha > 0\}$ is the cone of vector $\vec{v}$.

Linearity of representations requires that representations lie in a cone, but says nothing about their ordering. To model the structure of numeric properties, we introduce the constraint that the ordering of quantities is preserved in representation space.

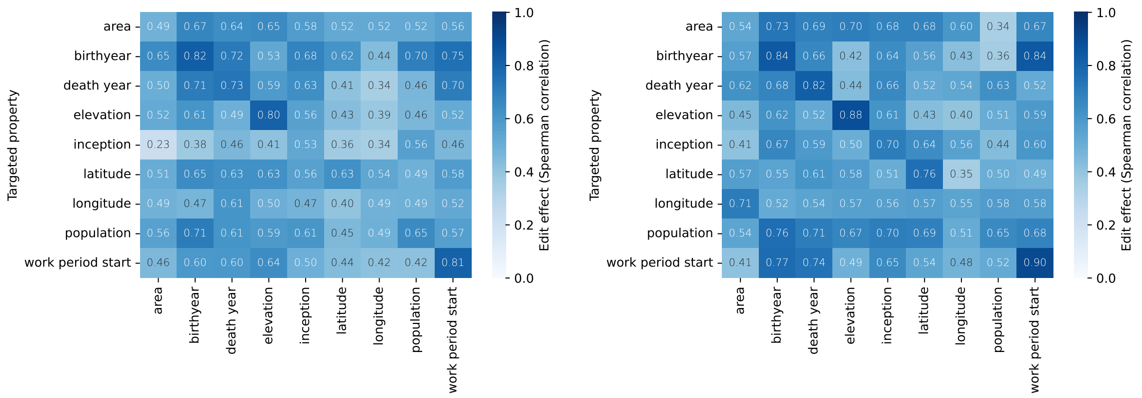
**Definition 2** (Monotonic representation of numeric properties)**.** A numeric property is represented monotonically if it is represented linearly in $\text{Cone}(\vec{u})$ and for all triples of attribute instances $h, i, j$ with quantities $q_h > q_i > q_j$ and representations $\vec{x_h}, \vec{x_i}, \vec{x_j}$ the following holds: $\vec{x_h} - \vec{x_j} = \alpha_{hj}\vec{u}$ and $\vec{x_i} - \vec{x_j} = \alpha_{ij}\vec{u}$ if and only if $\alpha_{hj} > \alpha_{ij}$.

There are many ways to operationalize this definition. One is to prepare a series of monotonic representations in $\text{Cone}(\vec{u})$ by varying $\alpha$ and then testing if these representations yield monotonic output changes, which is what we will do now.

**Method.** Viewing the LM as a causal graph (Meng et al., 2022; McGrath et al., 2023), we intervene via activation patching (Vig et al., 2020; Wang et al., 2022; Zhang and Nanda, 2024) and observe the effect on model output. Unlike the common setup in which one replaces activations from one input with activations from a different input, we patch activations along directions, similar to the manipulation method of Matsumoto et al. (2022).

Specifically, for each of the top $K$ directions $\vec{u_k} \in R^d, k \in [1\,..\,K]$ found by PLS, we prepare patches $\vec{p}_{s,k} = \alpha_s\vec{u_k}$ with edit weights $\alpha_s$ and step index $s \in [1\,..\,80]$. Lacking a principled method for choosing edit weights $\alpha_s$, we set their range to the minimum and maximum PLS loadings on each property's training split. This choice yields patches covering the empirical range of activation projections onto direction $\vec{u_k}$. After sampling $n_{train} = 1000$ popular entities for each of the six numeric properties we first fit PLS models for each property, then apply activation patches $\vec{p}_{s,k}$ to the representations of $n_{test} = 100$ held-out entities and for each entity record the LM's expressed quantity $y_{s,k}$. To evaluate monotonicity, i.e., the notion that small (large) edit weights $\alpha_s$ should have a small (large) effects and that negative (positive) weights should decrease (increase) the expressed quantity $y_{s,k}$, we quantify the intervention effect via the ranked Spearman correlation $\rho(\alpha_{s,k}; y_{s,k})$.

**Results.** We are interested in the effects and side effects on model output when patching activations along property-specific directions. Looking at effects first, we plot mean effects of directed activation patching across six numeric properties in Fig. 4. We see that there are properties




(a) Llama 2 7B (b) Llama 2 13B

Figure 5: Effects and side effects of directed activation patching. Diagonal entries (top-left to bottom right) show the effect on the targeted property in terms of mean Spearman correlation between edit weight $alpha_s, k$ and expressed quantity $y_s, k$. For example, patching an entity representation along a "birthyear" direction results in a corresponding change in the quantity expressed by Llama 2 13B with a correlation of $0.84$. Off-diagonal entries show side-effects, e.g., "birthyear" patches affect LM output when queried for an entity's death year with a correlation of $0.68$.

for which directed activation patching has highly monotonic effects, e.g., birthyear ($\rho = 0.84$), elevation ($\rho = 0.88$), or work period start ($\rho = 0.90$), suggesting that these properties have highly monotonic representations. Other properties exhibit a smaller degree of monotonic editability, e.g., longitude ($\rho = 0.55$) and population (0.65), suggesting that LM representations do not encode these properties as well. Figures for other models (see §G) lead to similar conclusions.

Having observed the effects of our interventions we now turn to their side effects on the expression of properties that were not the target of the intervention. For example, if we fitted a PLS regression to find "birthyear" directions, birthyear is our targeted property and all other properties, such as death year or longitude are non-targeted properties. Using the directions found in §2, we prompt LMs for non-targeted attributes, perform activation patching with weight $\alpha_s$ along a direction found for the targeted property and record expressed quantities $y'_{s,k}$. To see if non-targeted properties are affected in a similar monotonic fashion as targeted ones, we quantify the side-effect of directed activation patching as the mean Spearman correlation $\rho(\alpha_s, y'_{s,k})$, taken over 100 entities per property. We perform this procedure for all combinations of targeted and non-targeted properties, including three additional properties, and show results in Fig. 5. In this figure, diagonal entries show the mean effect on targeted properties and off-diagonal entries the size of side-effects. For Llama 2 7B, the mean effect size $\bar{\rho} = 0.65 \pm 0.12$ (mean of diagonal entries), is not much larger than the mean side-effect size $\bar{\rho} = 0.53 \pm 0.11$ (mean of off-diagonal entries). In contrast, for Llama 2 13B the effect size of $\bar{\rho} = 0.85 \pm 0.07$ is much larger than the size of side effects ($\bar{\rho} = 0.58 \pm 0.18$). A plausible explanation is that in Llama 2 7B properties share a subspace which encodes generic numeric or small-large ranges that are mapped to specific quantities depending on context, while the representation space of Llama 2 13B is more akin to a mixture of generic-numeric and property-specific subspaces. More work is needed to test this hypothesis.

The analysis of side-effects is complicated by real correlations between properties: Birthyear and death year distances are bounded by the human life span, latitude and population are correlated since the Earth's northern hemisphere is more populous, etc. Consequently, one might argue that, say, editing an entity's birthyear should also affect LM output when querying the entity's death year.

## 4 Conclusions

We used partial least-squares regression to identify low-dimensional subspaces of activation space that are predictive of the quantity an LM expresses when queried for numeric attributes such as an entity's birthyear. We then performed activation patching along directions in these subspaces and observed corresponding changes in model output. Our results suggest that LMs learn monotonic representations of numeric properties and that these representations exist in all of the examined LMs.

## 5 Limitations

### 5.1 General limitations of representational analysis

None of the language models studied in this work are embodied agents or otherwise capable of embodied cognition. Lacking direct sensorimotor grounding (Harnad, 1990; Mollo and Millière, 2023; Harnad, 2024), LMs cannot directly perceive, let alone precisely measure, the numerical attributes of which we claim to have found monotonic representations. It follows that any such representations are an artifact of distributional patterns in their training data, and that the best one can hope for is isomorphy between model representations and the properties of the real-world entities to which we tie those representations.

Leaving the groundedness of representations aside, the idea that concepts, knowledge, or behavior are "encoded" in neural representations might seem intuitively appealing, but has been strongly criticized, on theoretical grounds in the context of biological and artificial neural networks in general (Brette, 2019), and on empirical grounds in the context of pretrained language models in particular (Hase et al., 2023a; Niu et al., 2024).

Analysis of LM representations also has well-known limitations. Under the mild assumption that there exists a bijection between inputs and their representations, all information extractable from the input, i.e., the natural language prompt, can also be extracted from the LM's representation of that sequence (Pimentel et al., 2020b). Hence the question to be answered by representational analysis is not whether a feature of interest can be extracted or not, but how easy it is to extract. How to best quantify "ease of extraction" (Pimentel et al., 2020b) is an open question, although methods have been proposed (Pimentel et al., 2020a; Voita and Titov, 2020).

### 5.2 Specific limitations of the representational analysis conducted in this work

The low-dimensional linear subspaces found in this work allow relatively "easy" extraction when compared to the nominally high dimensionalities of activation space, but are still higher-dimensional than necessary, since the represented structures (e.g., years, geographic coordinates) are canonically one- to two-dimensional. Furthermore, activation space is nominally high-dimensional but its intrinsic dimension is believed to be much lower (Li et al., 2018; Aghajanyan et al., 2021; Razzhigaev et al., 2024). For example Razzhigaev et al. (2024) provide estimates for the intrinsic dimension of various LMs, ranging from about 10 to 70 dimensions (the models used in our experiments are not covered). If we view a non-linear, non-monotonic representation of full intrinsic dimensionality as the most complex encoding with worst-case ease of extraction, and one- to two-dimensional linear monotonic encodings as the simplest representation with optimal ease of extraction, then the low-dimensional subspaces we found fall somewhere between these bounds. Whether they are low-dimensional relative to the models' intrinsic dimension is currently unknown. Put differently, if the intrinsic dimension of Llama 2 7B turns out to be, say, 10, then finding, a 10-dimensional subspace that encodes all latitude information (see §D) is not surprising, but necessary.

While we found evidence for monotonic representation of numeric properties, it is likely that our causal interventions via activation patching along one-dimensional directions are too simplistic, considering the fact that according to our PLS regression results, numeric properties are encoded in low- but not one-dimensional subspaces. Hence it is possible that a more refined editing method operating on higher-dimensional directions will allow more precise control over LM output. Furthermore, our analysis is limited to popular entities, frequent numeric properties, and English queries, i.e., the combination most likely to be well-represented in the LM training data.

**Acknowledgements.** This work was supported by JST CREST Grant Number JPMJCR20D2 and JSPS KAKENHI Grant Number 21K17814.

## References


Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, Online. Association for Computational Linguistics.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models.

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Romain Brette. 2019. Is coding a relevant metaphor for the brain? *Behavioral and Brain Sciences*, 42:e215.

Roger C. Conant and W. Ross Ashby. 1970. Every good regulator of a system must be a model of that system. *International journal of systems science*, 1(2):89–97.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition.

Fahim Faisal and Antonios Anastasopoulos. 2023. Geographic and geopolitical biases of language models. In *Proceedings of the 3rd Workshop on Multi-lingual Representation Learning (MRL)*, pages 139–163, Singapore. Association for Computational Linguistics.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, Singapore. Association for Computational Linguistics.

Nathan Godey, Éric de la Clergerie, and Benoît Sagot. 2024. On the scaling laws of geographical representation in language models.

Abhijeet Gupta, Gemma Boleda, Marco Baroni, and Sebastian Padó. 2015. Distributional vectors encode referential attributes. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 12–21, Lisbon, Portugal. Association for Computational Linguistics.

Wes Gurnee and Max Tegmark. 2023. Language models represent space and time.

Stevan Harnad. 1990. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346.

Stevan Harnad. 2024. Language writ large: Llms, chatgpt, grounding, meaning and understanding.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. 2023a. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models.

Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. 2023b. Methods for measuring, updating, and visualizing factual beliefs in language models. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2714–2731, Dubrovnik, Croatia. Association for Computational Linguistics.

Benjamin Heinzerling and Kentaro Inui. 2021. Language models as knowledge bases: On entity representations, storage capacity, and paraphrased queries. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1772–1791, Online. Association for Computational Linguistics.

Benjamin Heinzerling, Michael Strube, and Chin-Yew Lin. 2017. Trust, but verify! better entity linking through automatic verification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 828–838, Valencia, Spain. Association for Computational Linguistics.

Evan Hernandez, Belinda Z. Li, and Jacob Andreas. 2023. Inspecting and editing knowledge representations in language models. In *Arxiv*.

J. D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Yibo Jiang, Goutham Rajendran, Pradeep Ravikumar, Bryon Aragam, and Victor Veitch. 2024. On the origins of linear representations in large language models.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Nora Kassner, Philipp Dufter, and Hinrich Schütze. 2021. Multilingual LAMA: Investigating knowledge in multilingual pretrained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3250–3258, Online. Association for Computational Linguistics.

János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. 2024. Atp*: An efficient and scalable method for localizing llm behaviour to components.

Karim Lasri, Tiago Pimentel, Alessandro Lenci, Thierry Poibeau, and Ryan Cotterell. 2022. Probing for the usage of grammatical number. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8818–8831, Dublin, Ireland. Association for Computational Linguistics.

Harvey Lederman and Kyle Mahowald. 2024. Are language models more like libraries or like librarians? bibliotechnism, the novel reference problem, and the attitudes of llms.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. 2018. Measuring the intrinsic dimension of objective landscapes. *arXiv preprint arXiv:1804.08838*.

Bastien Liétard, Mostafa Abdou, and Anders Søgaard. 2021. Do language models know the way to Rome? In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 510–517, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ziming Liu, Ouail Kitouni, Niklas S Nolte, Eric Michaud, Max Tegmark, and Mike Williams. 2022. Towards understanding grokking: An effective theory of representation learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 34651–34663. Curran Associates, Inc.

Max M. Louwerse and Rolf A. Zwaan. 2009. Language encodes geographical information. *Cognitive Science*, 33(1):51–73.

Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets.

Yuta Matsumoto, Benjamin Heinzerling, Masashi Yoshikawa, and Kentaro Inui. 2022. Tracing and manipulating intermediate values in neural math problem solvers. In *Proceedings of the 1st Workshop on Mathematical Natural Language Processing (MathNLP)*, pages 1–6, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Thomas McGrath, Matthew Rahtz, Janos Kramar, Vladimir Mikulik, and Shane Legg. 2023. The hydra effect: Emergent self-repair in language model computations.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems*, 36.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2023. A mechanism for solving relational tasks in transformer language models.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2021. Fast model editing at scale. *CoRR*.

Dimitri Coelho Mollo and Raphaël Millière. 2023. The vector grounding problem.

Giovanni Monea, Maxime Peyrard, Martin Josifoski, Vishrav Chaudhary, Jason Eisner, Emre Kıcıman, Hamid Palangi, Barun Patra, and Robert West. 2024. A glitch in the matrix? locating and detecting language model grounding with fakepedia.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 16–30, Singapore. Association for Computational Linguistics.

Jingcheng Niu, Andrew Liu, Zining Zhu, and Gerald Penn. 2024. What does the knowledge neuron thesis have to do with knowledge? In *The Twelfth International Conference on Learning Representations*.

The Pandas development team. 2020. pandas-dev/pandas: Pandas.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2023. The linear representation hypothesis and the geometry of large language models.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Karl Pearson. 1901. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

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, Hong Kong, China. Association for Computational Linguistics.

Tiago Pimentel, Naomi Saphra, Adina Williams, and Ryan Cotterell. 2020a. Pareto probing: Trading off accuracy for complexity. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3138–3153, Online. Association for Computational Linguistics.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. 2020b. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622, Online. Association for Computational Linguistics.

Ben Prystawski, Michael Y. Li, and Noah D. Goodman. 2023. Why think step by step? reasoning emerges from the locality of experience.

Anton Razzhigaev, Matvey Mikhalchuk, Elizaveta Goncharova, Ivan Oseledets, Denis Dimitrov, and Andrey Kuznetsov. 2024. The shape of learning: Anisotropy and intrinsic dimensions in transformer-based models.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Cody Rushing and Neel Nanda. 2024. Explorations of self-repair in language models.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Curt Tigges, Oskar John Hollinsworth, Atticus Geiger, and Neel Nanda. 2023. Linear representations of sentiment in large language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. 2020. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Elena Voita and Ivan Titov. 2020. Information-theoretic probing with minimum description length. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 183–196, Online. Association for Computational Linguistics.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small.

Michael L. Waskom. 2021. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

Svante Wold, Michael Sjöström, and Lennart Eriksson. 2001. Pls-regression: a basic tool of chemometrics. *Chemometrics and intelligent laboratory systems*, 58(2):109–130.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Paul Youssef, Osman Koraş, Meijie Li, Jörg Schlötterer, and Christin Seifert. 2023. Give me the facts! a survey on factual knowledge probing in pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15588–15605, Singapore. Association for Computational Linguistics.

Fred Zhang and Neel Nanda. 2024. Towards best practices of activation patching in language models: Metrics and methods.

Zexuan Zhong, Dan Friedman, and Danqi Chen. 2021. Factual probing is [MASK]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5017–5033, Online. Association for Computational Linguistics.

## A Additional related work

Shaped by the locality of physical reality, the locality of human experience (Prystawski et al., 2023) gives rise to distributional patterns of language use. Such patterns include patterns of geographic and temporal coherence (Heinzerling et al., 2017), which reflect spatiotemporal proximity of real-world entities. These patterns can be picked up by statistical models and allow, e.g., to predict geographic information from co-occurrence statistics of cities mentioned in news articles (Louwerse and Zwaan, 2009). Probing static word vector representations for numeric attributes of geopolitical entities, Gupta et al. (2015) obtain good relative rankings, but do not evaluate absolute values nor analyze the geometry of representations. Continuing this line of research, Liétard et al. (2021) probe LM representations for GPS coordinates. Perhaps due to the—by current standards—small scale of the studied LMs, they find only limited success but report that larger models appeared to encode more geographic information. Faisal and Anastasopoulos (2023) measure how well the geographic proximity of countries can be recovered from LM representations but differ from our work in their focus on the impact of politico-cultural factors.

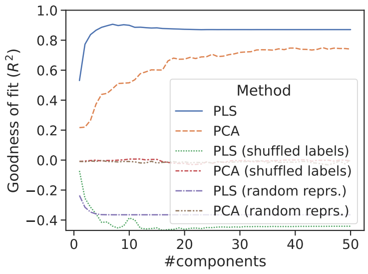
Closest to our work is the analysis of geo-temporal information encoded in Llama 2 representations by Gurnee and Tegmark (2023). Our work corroborates their finding of linear subspaces of activation space which are predictive of numeric attributes, but is distinct in three important aspects. First, as we show in §2, the subspaces found PCA, as used by Gurnee and Tegmark, are of considerably higher dimensionality ($50 - 100$) than the subspaces found by partial least-square regression ($2 - 17$). Our finding thus tightens the upper bound on the complexity of numeric property representation in recent LMs. Second, we make explicit and formalize the notion of monotonic representation. Third, our interventions via directed activation patching (§3) found one-dimensional directions with fine-grained effects on the expression of numeric attributes, across all numeric properties and models we analyzed, thereby establishing a causal relationship between monotonic representations and LM behavior.

# B Data sample

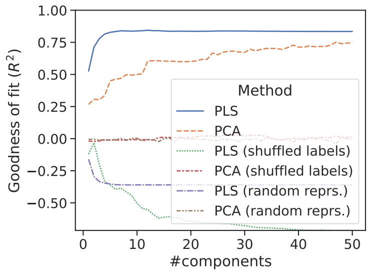
| Property | Prop. ID | Entity | Entity ID | Prompt | Value | Unit |
|---|---|---|---|---|---|---|
| birthyear | P569 | Nina Foch | Q235632 | In what year was Nina Foch born? | 1924 | annum |
| birthyear | P569 | Geoffrey Holder | Q945691 | In what year was Geoffrey Holder born? | 1930 | annum |
| birthyear | P569 | Harriette L. Chandler | Q5664432 | In what year was Harriette L. Chandler born? | 1937 | annum |
| birthyear | P569 | Gabriel García Márquez | Q5878 | In what year was Gabriel García Márquez born? | 1927 | annum |
| birthyear | P569 | Norman Schwarzkopf Jr. | Q310188 | In what year was Norman Schwarzkopf Jr. born? | 1934 | annum |
| birthyear | P569 | Paul de Vos | Q2610964 | In what year was Paul de Vos born? | 1590 | annum |
| birthyear | P569 | Nicolas Carnot | Q181685 | In what year was Nicolas Carnot born? | 1796 | annum |
| birthyear | P569 | Steve Harvey | Q2347009 | In what year was Steve Harvey born? | 1957 | annum |
| birthyear | P569 | Tommy Lawton | Q726272 | In what year was Tommy Lawton born? | 1919 | annum |
| birthyear | P569 | Hans von Bülow | Q155540 | In what year was Hans von Bülow born? | 1830 | annum |
| death year | P570 | Johannes R. Becher | Q58057 | In what year did Johannes R. Becher die? | 1958 | annum |
| death year | P570 | Friedrich Georg Wilhelm von Struve | Q57164 | In what year did Friedrich Georg Wilhelm von Struve die? | 1864 | annum |
| death year | P570 | Pierre Boulez | Q156193 | In what year did Pierre Boulez die? | 2016 | annum |
| death year | P570 | Giovanni da Palestrina | Q179277 | In what year did Giovanni da Palestrina die? | 1594 | annum |
| death year | P570 | Abdurrauf Fitrat | Q317907 | In what year did Abdurrauf Fitrat die? | 1938 | annum |
| death year | P570 | Lucian Freud | Q154594 | In what year did Lucian Freud die? | 2011 | annum |
| death year | P570 | Akseli Gallen-Kallela | Q170068 | In what year did Akseli Gallen-Kallela die? | 1931 | annum |
| death year | P570 | Spock | Q16341 | In what year did Spock die? | 2263 | annum |
| death year | P570 | William Orpen | Q922483 | In what year did William Orpen die? | 1931 | annum |
| death year | P570 | Carlos Santiago Mérida | Q1043100 | In what year did Carlos Santiago Mérida die? | 1984 | annum |
| population | P1082 | Akhisar | Q209905 | What is the population of Akhisar? | 173026 | 1 |
| population | P1082 | Tripura | Q1363 | What is the population of Tripura? | 3665958 | 1 |
| population | P1082 | Albert | Q30940 | What is the population of Albert? | 9930 | 1 |
| population | P1082 | High Wycombe | Q64116 | What is the population of High Wycombe? | 120256 | 1 |
| population | P1082 | Plön | Q497060 | What is the population of Plön? | 8914 | 1 |
| population | P1082 | Republika Srpska | Q11196 | What is the population of Republika Srpska? | 1228423 | 1 |
| population | P1082 | Lebanese | Q2606511 | What is the population of Lebanese? | 8000000 | 1 |
| population | P1082 | Geraardsbergen | Q499532 | What is the population of Geraardsbergen? | 33403 | 1 |
| population | P1082 | Gorzów Wielkopolski | Q104731 | What is the population of Gorzów Wielkopolski? | 124295 | 1 |
| population | P1082 | Harran | Q199547 | What is the population of Harran? | 47606 | 1 |
| evelation | P2044 | Sondrio | Q6274 | How high is Sondrio? | 360 | metre |
| evelation | P2044 | Rio Branco | Q171612 | How high is Rio Branco? | 158 | metre |
| evelation | P2044 | Demmin | Q50960 | How high is Demmin? | 8 | metre |
| evelation | P2044 | Cetinje | Q173338 | How high is Cetinje? | 650 | metre |
| evelation | P2044 | Highland Park | Q576671 | How high is Highland Park? | 503 | metre |
| evelation | P2044 | Gozo | Q170488 | How high is Gozo? | 195 | metre |
| evelation | P2044 | Saint-Jean-de-Maurienne | Q208860 | How high is Saint-Jean-de-Maurienne? | 566 | metre |
| evelation | P2044 | Butte | Q467664 | How high is Butte? | 1688 | metre |
| evelation | P2044 | Cottbus | Q3214 | How high is Cottbus? | 76 | metre |
| evelation | P2044 | Mahilioŭ Region | Q189822 | How high is Mahilioŭ Region? | 191 | metre |
| longitude | P625.long | Korean Empire | Q28233 | What is the longitude of Korean Empire? | 126.98 | degree |
| longitude | P625.long | Pine Bluff | Q80012 | What is the longitude of Pine Bluff? | -92.00 | degree |
| longitude | P625.long | Tegernsee | Q260130 | What is the longitude of Tegernsee? | 11.76 | degree |
| longitude | P625.long | Old Cölln | Q269622 | What is the longitude of Old Cölln? | 13.40 | degree |
| longitude | P625.long | Cambridge | Q49111 | What is the longitude of Cambridge? | -71.11 | degree |
| longitude | P625.long | Stryn | Q5223 | What is the longitude of Stryn? | 6.86 | degree |
| longitude | P625.long | Ciudad Real Province | Q54932 | What is the longitude of Ciudad Real Province? | -4.00 | degree |
| longitude | P625.long | Santa Catarina | Q41115 | What is the longitude of Santa Catarina? | -50.49 | degree |
| longitude | P625.long | Wake Forest University | Q392667 | What is the longitude of Wake Forest University? | -80.28 | degree |
| longitude | P625.long | West Lothian | Q204940 | What is the longitude of West Lothian? | -3.50 | degree |
| latitude | P625.lat | Küsnacht | Q69216 | What is the latitude of Küsnacht? | 47.32 | degree |
| latitude | P625.lat | Mount Jerome Cemetery | Q917854 | What is the latitude of Mount Jerome Cemetery? | 53.32 | degree |
| latitude | P625.lat | Dayton Children’s Hospital | Q5243510 | What is the latitude of Dayton Children’s Hospital? | 39.77 | degree |
| latitude | P625.lat | Le Flore County | Q495944 | What is the latitude of Le Flore County? | 34.90 | degree |
| latitude | P625.lat | Czechoslovakia | Q33946 | What is the latitude of Czechoslovakia? | 50.08 | degree |
| latitude | P625.lat | Pembroke College | Q956501 | What is the latitude of Pembroke College? | 52.20 | degree |
| latitude | P625.lat | Hayward | Q491114 | What is the latitude of Hayward? | 37.67 | degree |
| latitude | P625.lat | Banaskantha district | Q806125 | What is the latitude of Banaskantha district? | 24.17 | degree |
| latitude | P625.lat | Corbeil-Essonnes | Q208812 | What is the latitude of Corbeil-Essonnes? | 48.61 | degree |
| latitude | P625.lat | Elbasan | Q114257 | What is the latitude of Elbasan? | 41.11 | degree |

Table 1: Random sample of the entities used in our experiments, along with corresponding numeric attributes and prompts. Entities, their English labels, and numeric attributes for each property are extracted from an April 2022 dump of Wikidata (`wikidata-20220421-all`). In many cases Wikidata contains multiple values for a given numeric attribute, e.g., reflecting chronological change such as the population of a city, or owing to conflicting sources. In such cases we take the mode of the distribution as gold value. We also filter out quantities with non-standard units, such as elevations measured in feet.

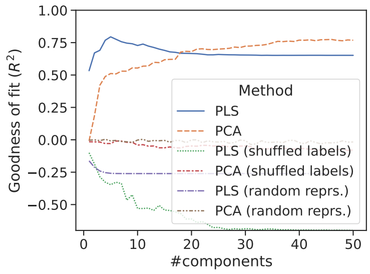
# C Regression on entity representations: Additional figures



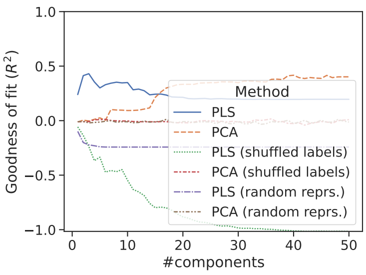

(c) Birthyear



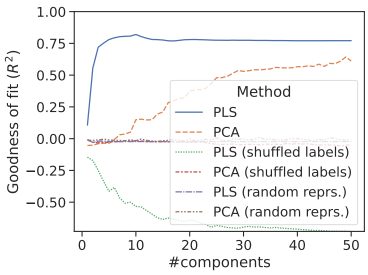

(d) Death year



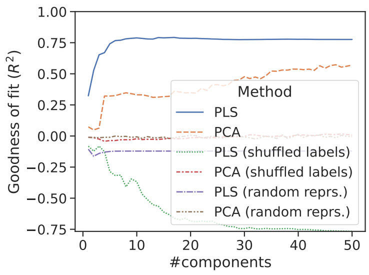

(e) Population




(f) Elevation

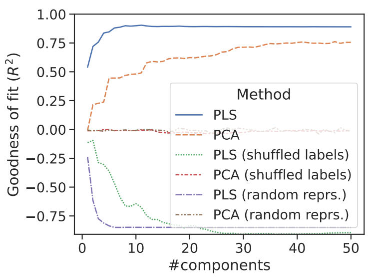



(g) Latitude

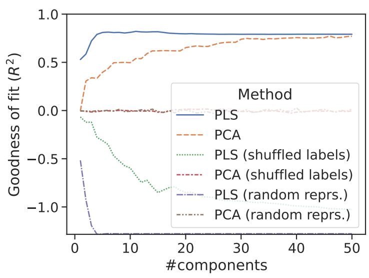



(h) Longitude

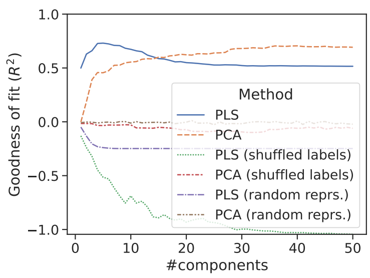
Figure 6: Low-dimensional subspaces of Llama-2-13B's 5120-dimensional activation space are predictive of the quantity expressed by the LM when queried for a numeric attribute of an entity, across six different numeric properties. Each subfigure shows the performance of a regression model fitted to predict the expressed quantities from LM-internal entity representations (in layer $l = 0.3$), as a function of the number of PCA/PLS components used for prediction. Unlike regression on PCA components (dashed orange), partial least squares regression (PLS, solid blue) identifies a small set of predictive components. Controls with shuffled labels (dotted green, dash-dotted red) and random entity representations (long-dash-dot purple, dash-dot-dot brown) fail to find predictive subspaces.



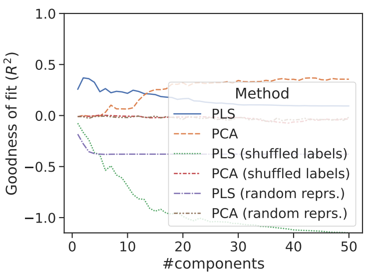

(a) Birthyear



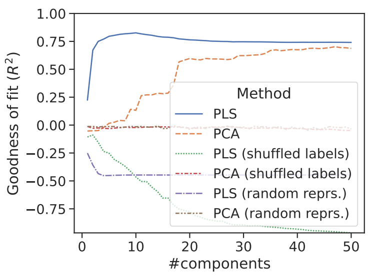

(b) Death year



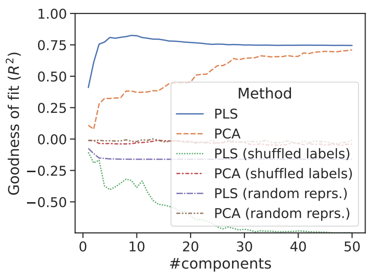

(c) Population




(d) Elevation

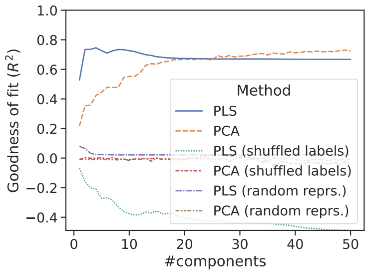



(e) Latitude

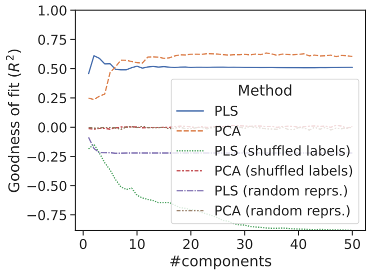



(f) Longitude

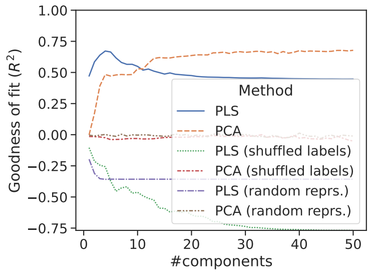
Figure 7: Regression curves for Llama 2 7B. See explanation in Fig. 6.



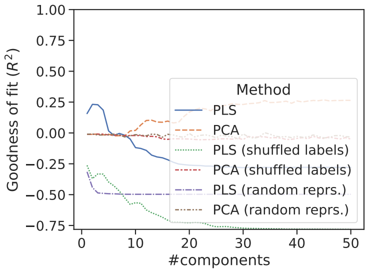

(a) Birthyear



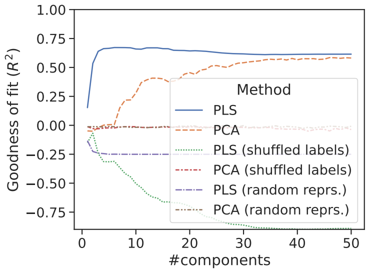

(b) Death year



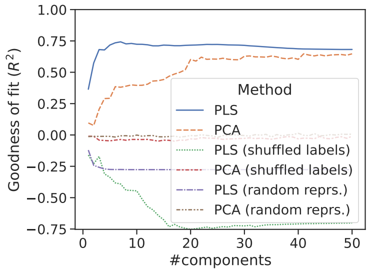

(c) Population




(d) Elevation

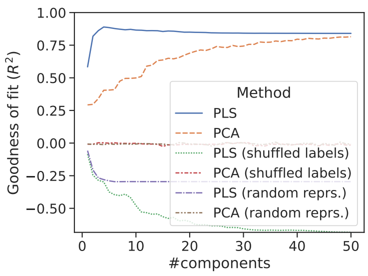



(e) Latitude

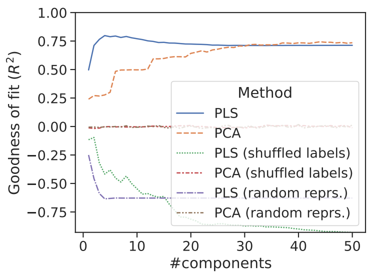



(f) Longitude

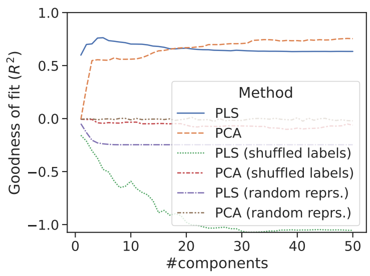
Figure 8: Regression curves for Falcon 7B. See explanation in Fig. 6.



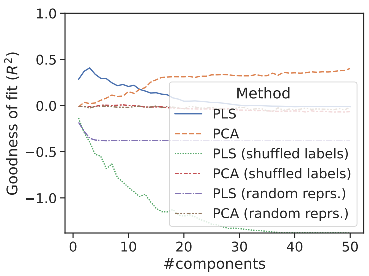

(a) Birthyear



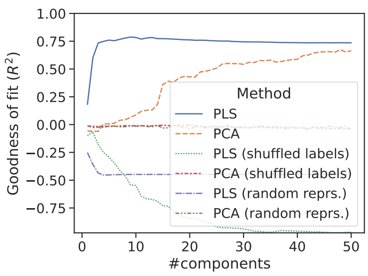

(b) Death year



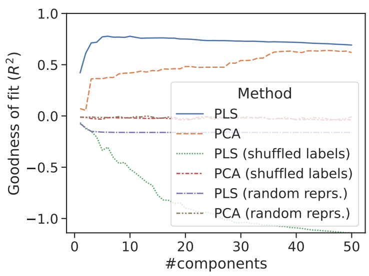

(c) Population




(d) Elevation




(e) Latitude




(f) Longitude

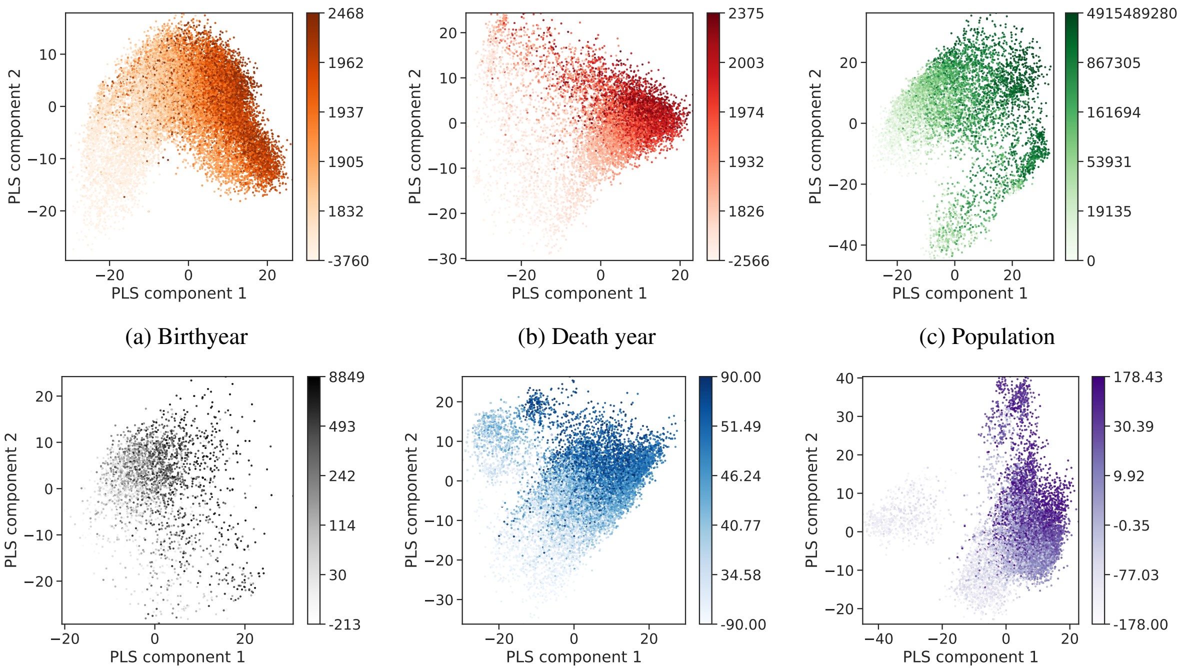
Figure 9: Regression curves for Mistral 7B. See explanation in Fig. 6.

## D Regression on entity representations: Additional analysis

| Property | Model | $R^2$ | $C\,[maxR^2]$ | $C\,[\geq 0.95R^2]$ | $C\,[\geq 0.90R^2]$ | $C\,[\geq 0.80R^2]$ | $C\,[\geq 0.70R^2]$ | $C\,[\geq 0.60R^2]$ | $C\,[\geq 0.50R^2]$ |
|---|---|---|---|---|---|---|---|---|---|
| birthyear (P569) | Falcon 7B | 0.75 | 4 | 2 | 2 | 2 | 1 | 1 | 1 |
| birthyear (P569) | Llama 2 13B | 0.91 | 7 | 4 | 3 | 2 | 2 | 2 | 1 |
| birthyear (P569) | Llama 2 7B | 0.90 | 11 | 6 | 4 | 3 | 2 | 2 | 1 |
| birthyear (P569) | Mistral 7B | 0.89 | 4 | 3 | 2 | 2 | 2 | 1 | 1 |
| death year (P570) | Falcon 7B | 0.61 | 2 | 2 | 2 | 2 | 1 | 1 | 1 |
| death year (P570) | Llama 2 13B | 0.84 | 12 | 4 | 3 | 2 | 2 | 1 | 1 |
| death year (P570) | Llama 2 7B | 0.82 | 11 | 4 | 4 | 3 | 2 | 1 | 1 |
| death year (P570) | Mistral 7B | 0.80 | 4 | 3 | 3 | 2 | 2 | 1 | 1 |
| latitude (P625.lat) | Falcon 7B | 0.67 | 6 | 3 | 3 | 3 | 2 | 2 | 2 |
| latitude (P625.lat) | Llama 2 13B | 0.82 | 10 | 5 | 4 | 3 | 3 | 2 | 2 |
| latitude (P625.lat) | Llama 2 7B | 0.83 | 10 | 5 | 3 | 2 | 2 | 2 | 2 |
| latitude (P625.lat) | Mistral 7B | 0.79 | 9 | 4 | 3 | 3 | 2 | 2 | 2 |
| longitude (P625.long) | Falcon 7B | 0.74 | 7 | 5 | 3 | 3 | 2 | 2 | 2 |
| longitude (P625.long) | Llama 2 13B | 0.79 | 17 | 6 | 5 | 3 | 3 | 2 | 2 |
| longitude (P625.long) | Llama 2 7B | 0.83 | 9 | 5 | 3 | 3 | 2 | 2 | 2 |
| longitude (P625.long) | Mistral 7B | 0.78 | 6 | 5 | 3 | 3 | 2 | 2 | 1 |
| population (P1082) | Falcon 7B | 0.67 | 4 | 3 | 3 | 2 | 1 | 1 | 1 |
| population (P1082) | Llama 2 13B | 0.79 | 5 | 4 | 4 | 2 | 2 | 1 | 1 |
| population (P1082) | Llama 2 7B | 0.73 | 5 | 4 | 3 | 2 | 2 | 1 | 1 |
| population (P1082) | Mistral 7B | 0.76 | 5 | 4 | 2 | 2 | 1 | 1 | 1 |
| elevation (P2044) | Falcon 7B | 0.23 | 2 | 2 | 2 | 2 | 2 | 1 | 1 |
| elevation (P2044) | Llama 2 13B | 0.43 | 3 | 2 | 2 | 2 | 2 | 2 | 1 |
| elevation (P2044) | Llama 2 7B | 0.37 | 2 | 2 | 2 | 2 | 2 | 1 | 1 |
| elevation (P2044) | Mistral 7B | 0.41 | 3 | 3 | 2 | 2 | 2 | 1 | 1 |

Table 2: Number of partial least squares regression components $C\,[T]$ required for a given goodness of fit $T$, found using the experimental setup described in §2. For example, the $C\left[\geq 0.95R^2\right]$ column shows the number of components required to reach 95 percent of the maximum goodness of fit for the respective property and model. From this column we can read that, e.g., two components of Falcon 7B's activation space are sufficient to reach 95 percent of the maximum goodness of fit when predicting the birthyear of entities, indicating that this property is almost entirely encoded in a two-dimensional subspace of this model's activation space.

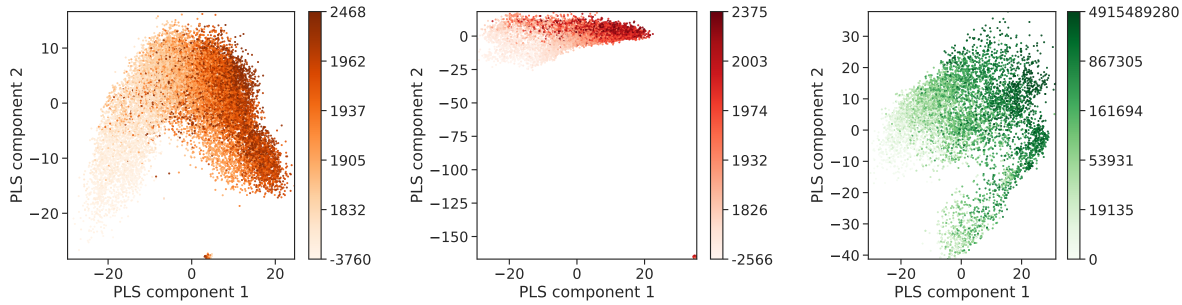
## E PLS projections of entity representations: Additional figures



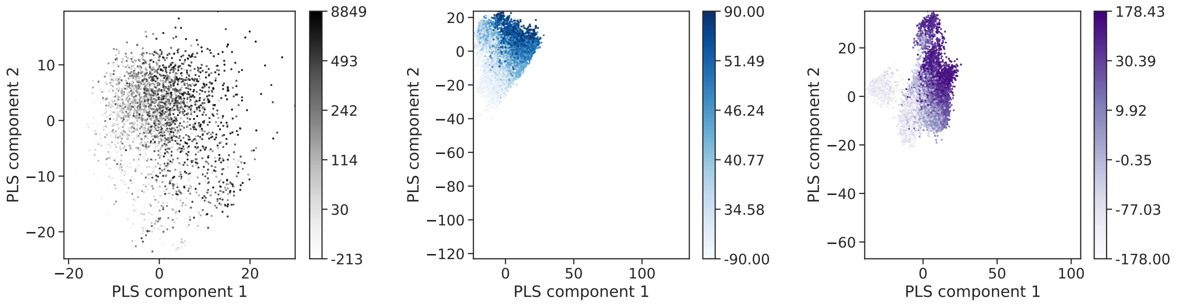

(a) Birthyear (b) Death year (c) Population

(d) Elevation (e) Latitude (f) Longitude

Figure 10: Projection onto the top two components of per-property partial least squares regressions reveals monotonic structure in LM representations. We first fit a PLS model on Llama 2 13B entity representations from our training split for each property, project entity representations from the test split, and then plot the resulting 2-d projections. Each dot represents one entity and color saturation represents the value of the corresponding entity attribute. See units for each property in Table 1.

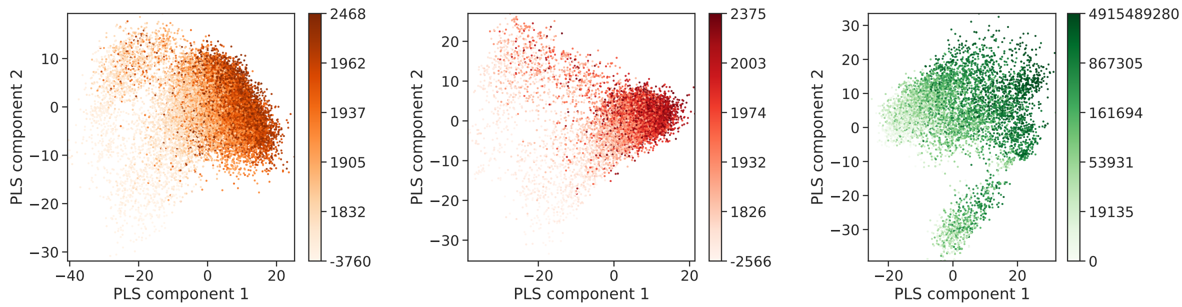



(a) Birthyear (b) Death year (c) Population

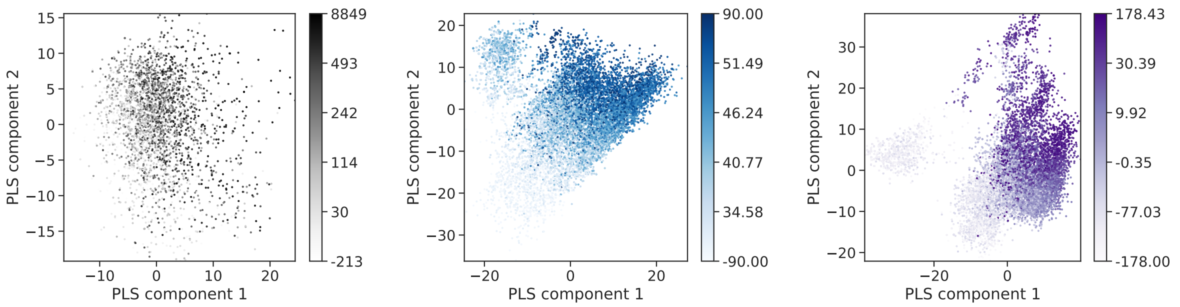



(d) Elevation (e) Latitude (f) Longitude

Figure 11: PLS projections of Llama 2 7B entity representations. See explanation in Fig. 10.

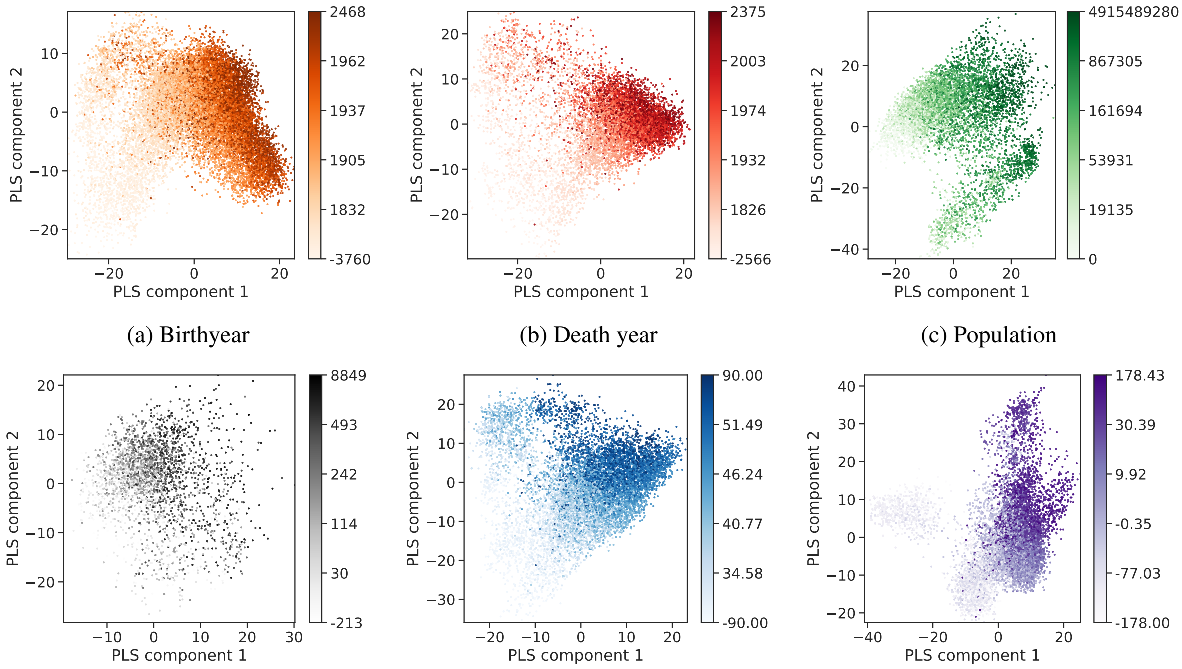



(a) Birthyear (b) Death year (c) Population




(d) Elevation (e) Latitude (f) Longitude

Figure 12: PLS projections of Falcon 7B entity representations. See explanation in Fig. 10.




(a) Birthyear (b) Death year (c) Population

(d) Elevation (e) Latitude (f) Longitude

Figure 13: PLS projections of Mistral 7B entity representations. See explanation in Fig. 10.

# F Choice of probing and edit locus

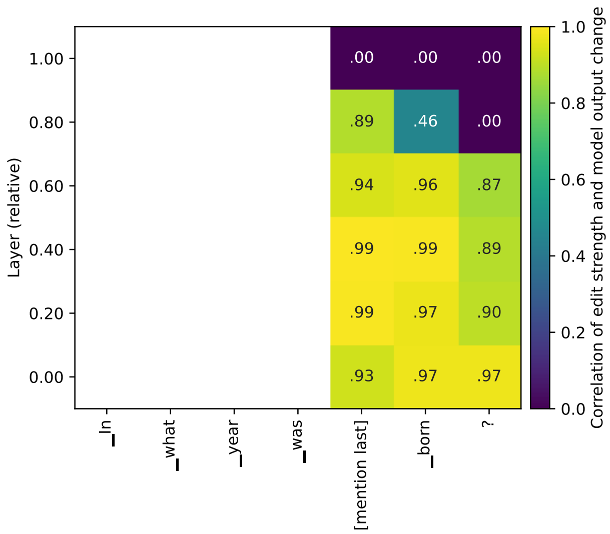


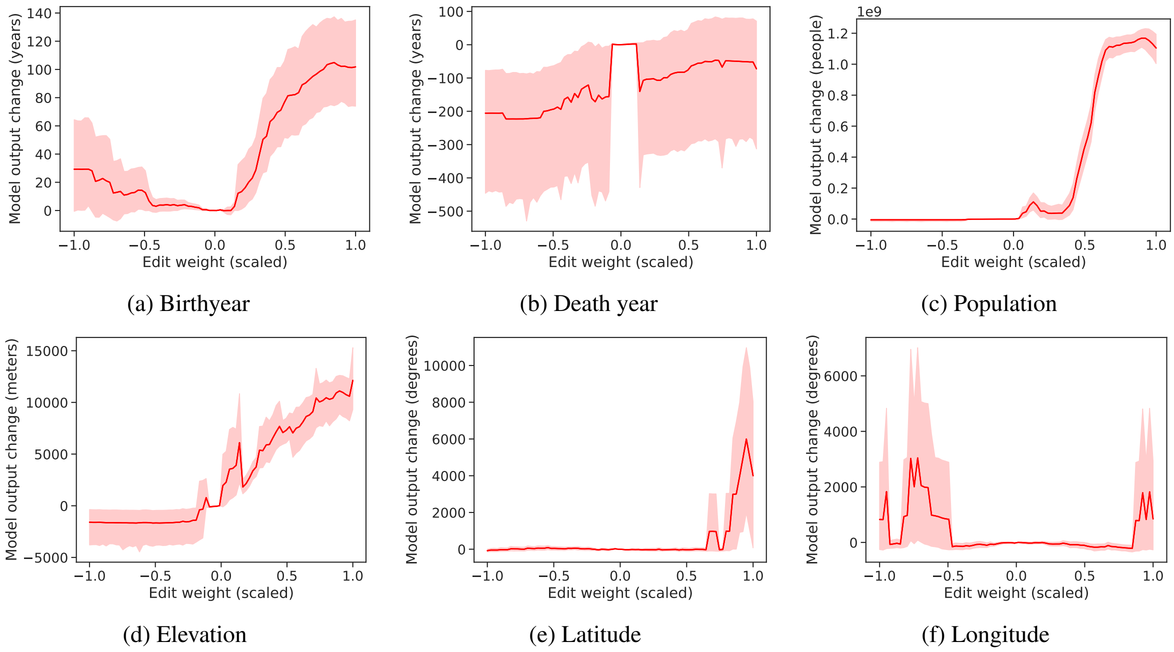

Figure 14: Results of a cursory search for the best probing and edit locus, using Llama 2 7B.

Varying token position and layer, we edit the hidden state at this locus as described in §3 and record the Spearman correlation between edit strength and the change in the quantity (here: birthyear) expressed by the model. Correlation is highest (0.99) in the region between layers 0.2 and 0.4 and the last subword token of the entity mention and the following token. Based on this, we choose the last mention token and the middle point at layer $l = 0.3$ as locus for the regression experiments in §2 and activation patching experiments in §3, across all numeric properties and LMs, but acknowledge that a more exhaustive search would likely find better probing and edit loci.

A question left open so far is where activation patching should be performed. While automatic methods for localizing model components and subnetworks of interest have been proposed (Conmy et al., 2023;

Kramár et al., 2024), for simplicity we perform a coarse search across layers and token positions for one numeric property and use the found setting for all experiments (see §F). In addition to this edit locus, we also search for an edit window, whose purpose is to counteract iterative inference effects (McGrath et al., 2023; Rushing and Nanda, 2024). Layer-wise we find that a window of $\pm 2$ layers around the edit locus is most effective, which is smaller than the $\pm 5$ layers used in prior work (Meng et al., 2022; Hase et al., 2023a). We also implement a token-wise window (Monea et al., 2024), finding that in addition to the last entity mention token, patching up to two token representations to the left and one token representation to the right works best for the prompts in our experiments. Typically, this token window size covers the entity mention and the main verb or last token of the prompt, depending on the numeric property (see prompts in §B). In summary, we patch activations in a 5-layer window centered on layer $l = 0.3$ and an up-to 4-token window surrounding the last entity mention token. To improve output format adherence, we append the instruction *One word answer only* to all prompts.

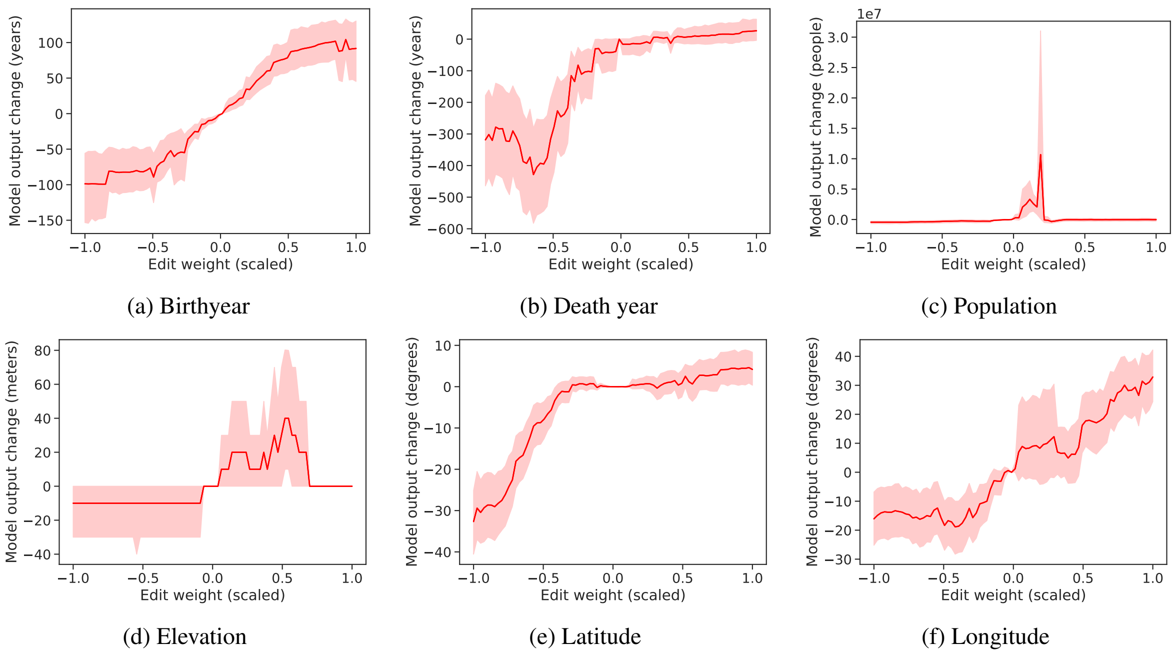
# G Edit curves for additional language models




(a) Birthyear (b) Death year (c) Population

(d) Elevation (e) Latitude (f) Longitude

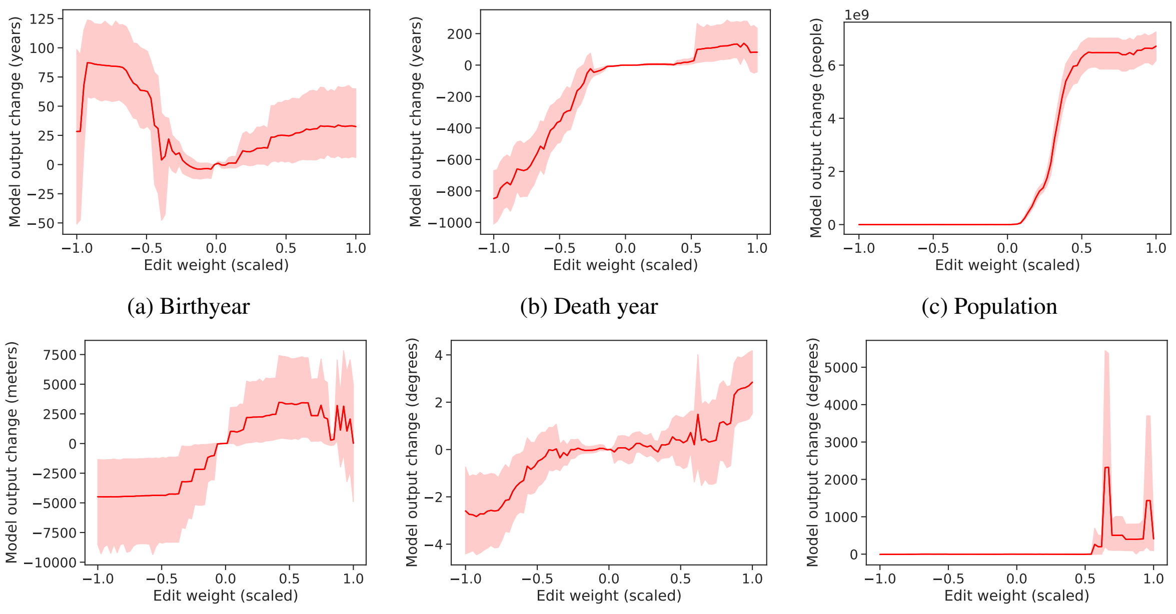
Figure 15: Effect of activation patching along property-specific directions across several numeric properties with Llama 2 7B. See explanation in Fig. 4.




(a) Birthyear (b) Death year (c) Population

(d) Elevation (e) Latitude (f) Longitude

Figure 16: Effect of activation patching along property-specific directions across several numeric properties with Falcon 7B (Almazrouei et al., 2023). See explanation in Fig. 4.




(a) Birthyear (b) Death year (c) Population

(d) Elevation (e) Latitude (f) Longitude

Figure 17: Effect of activation patching along property-specific directions across several numeric properties with Mistral 7B (Jiang et al., 2023). See explanation in Fig. 4.

## H Effect of property-encoding directions: Model output examples

| $\alpha_s$ | $y_{s,1}$ | $y_{s,2}$ | $y_{s,3}$ | $y_{s,4}$ | $y_{s,5}$ | $y_{s,6}$ |
|---|---|---|---|---|---|---|
| 1.00 | 1941 | 1955 | 1980 | 1980 | 2012 | 1929 |
| 0.90 | 1941 | 1955 | 1955 | 1984 | 2012 | 1929 |
| 0.80 | 1941 | 1955 | 1955 | 1984 | 2012 | 1929 |
| 0.70 | 1941 | 1955 | 1955 | 1980 | 1968 | 1929 |
| 0.60 | 1932 | 1955 | 1935 | 1958 | 1968 | 1929 |
| 0.50 | 1932 | 1940 | 1935 | 1958 | 1964 | 1929 |
| 0.40 | 1932 | 1930 | 1917 | 1958 | 1957 | 1902 |
| 0.30 | 1929 | 1930 | 1906 | 1958 | 1929 | 1902 |
| 0.20 | 1902 | 1902 | 1902 | 1934 | 1929 | 1902 |
| 0.10 | 1902 | 1902 | 1902 | 1902 | 1902 | 1902 |
| 0.00 | 1902 | 1902 | 1902 | 1902 | 1902 | 1902 |
| -0.10 | 1887 | 1902 | 1902 | 1902 | 1882 | 1902 |
| -0.20 | 1882 | 1902 | 1902 | 1887 | 1882 | 1902 |
| -0.30 | 1883 | 1902 | 1902 | 1887 | 1882 | 1902 |
| -0.40 | 1619 | 1902 | 1906 | 1887 | 1882 | 1901 |
| -0.50 | 1619 | 1902 | 1906 | 1887 | 1882 | 1906 |
| -0.60 | 1619 | 1902 | 1906 | 1887 | 1882 | 1906 |
| -0.70 | 1619 | 1902 | 1906 | 1887 | 1880 | 1906 |
| -0.80 | 1888 | 1902 | 1902 | 1887 | 1880 | 1906 |
| -0.90 | 1815 | 1902 | 1902 | 1858 | 1880 | 1906 |
| -1.00 | 1815 | 1902 | 1902 | 1858 | 1880 | 1906 |
| $\rho(\alpha_s, y_{s,k})$ | 0.91 | 0.87 | 0.72 | 0.97 | **0.98** | 0.39 |

(a) Birthyear of Karl Popper

| $\alpha_s$ | $y_{s,1}$ |
|---|---|
| 1.00 | 7.5 billion |
| 0.90 | 7.5 billion |
| 0.80 | 7.5 billion |
| 0.70 | 7.5 billion |
| 0.60 | 7.5 billion |
| 0.50 | 7.5 billion |
| 0.40 | 1.3 billion |
| 0.30 | 1.3 billion |
| 0.20 | 1.3 billion |
| 0.10 | 10 million |
| 0.00 | 40,000 |
| -0.10 | 40,000 |
| -0.20 | 25,000 |
| -0.30 | 25,000 |
| -0.40 | 20,000 |
| -0.50 | 20,000 |
| -0.60 | 20,000 |
| -0.70 | 12,000 |
| -0.80 | 12,000 |
| -0.90 | 12,000 |
| -1.00 | 12,000 |
| $\rho(\alpha_s, y_{s,k})$ | **0.98** |

(b) Population of Zittau

Table 3: The quantity $y_{s,k}$ expressed by a LM changes as a result of directed activation patching along direction $k$ with (normalized) edit weight $\alpha_s$, with $\alpha_s = 0.00$ corresponding to unedited model activations. Warm colors indicate values larger than and cold colors values smaller than the true value, which, if output by the LM, is printed black. Table (a) shows how one-dimensional directed patches along each of the top six "birthyear" PLS components change the answer given by Llama 2 13B to the prompt: *In what year was Karl Popper born? One word answer only*. It is apparent that the most-correlated component ($k = 1$) does not necessarily correspond to the direction in which model behavior exhibits highest monotonicity, which in this case is component $k = 5$ with a Spearman correlation of 0.98. Table (b) shows the effect of patching along the top "population" component on Llama 2 13B when prompted: *What is the population of Zittau? One word answer only*.

Table 3 gives examples of how numeric attribute expression changes as a result of directed activation patching. Patching along "birthyear" directions results in the expression of different years, although the degree of monotonicity, as quantified by Spearman correlation $\rho$, varies. Patching along the top "population" direction causes the model to generate a range of outputs that can be interpreted as population sizes, although the largest values are more suited to a planetary than a municipal scale. The sequence of outputs has rather sudden jumps, e.g., from *40,000* (unedited model, $\alpha_s = 0.00$) to *10 million* after taking the first step in the "larger population" direction ($\alpha_s = 0.10$). The pattern of jumps and plateaus is plausibly connected to several factors such as tokenization effects and the likely high frequency of certain numerals (*1.3 billion*: population of China at some point in time; *7.5 billion*: population of Earth, etc.) in the training data, but we leave a detailed investigation to future work. The pattern also indicates that activation space, while apparently monotonic, is not linear in this direction. The intervention also induces a switch from positional notation (*40,000*) to named numbers (*million*, *billion*), which showcases effects beyond single tokens.

## I Software

The following is a list of the main libraries used in this work:

- Numpy (Harris et al., 2020)
- Scikit-learn (Pedregosa et al., 2011)
- Pytorch (Paszke et al., 2019)
- Transformers (Wolf et al., 2020)
- seaborn (Waskom, 2021)
- Matplotlib (Hunter, 2007)
- SciPy (Virtanen et al., 2020)
- Pandas (Pandas development team, 2020)

We thank all authors and the open source community in general for creating and maintaining publicly and freely available software.