Linear Representation of Numeric Attributes in Language Models

Anonymous ACL submission

Abstract

Language models (LMs) can express factual knowledge involving numeric attributes such as *Karl Popper was born in 1902*. However, if and how such knowledge is encoded in the model's internal representations is poorly understood. Here, we introduce a simple method for finding and editing representations of numeric attributes such as an entity's birth year. Empirically, we find low-dimensional subspaces that encode numeric attributes in an interpretable and easily editable fashion. When editing representations along directions in these subspaces, LM output changes accordingly. For example, by patching activations along a "birthyear" direction we can make the LM express an increasingly late birthyear, e.g., "Karl Popper was born in 1913", "Karl Popper was born in 1944", "Karl Popper was born in 1975". Attribute-encoding directions exist across several numeric properties in all models under consideration, suggesting the intruiging possibility that linear representations of numeric attributes reliably emerge during LM pretraining.

1 Introduction

Language models (LMs) are capable of expressing factual knowledge (Petroni et al., 2019; Jiang et al., 2020; Roberts et al., 2020; Heinzerling and Inui, 2021; Kassner et al., 2021). For example, Llama 2 (Touvron et al., 2023) correctly answers the question *In which year was Karl Popper born?* While the question if and to what degree LMs can be said to "know" anything at all is subject of ongoing debate (Bender and Koller, 2020; Hase et al., 2023; 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 so far mainly targeted entity-entity relations such as *Warsaw is the capital of Poland* (Merullo et al., 2023). If and how LM representations encode factual knowledge involving numerical attributes, such as an entity's birthyear, is less understood. Unlike entity-entity relational knowledge, numerical attributes have natural orderings and structure, such as earlier/later and smaller/larger scales or geographic coordinate systems. While this kind of structure is natural and intuitive for humans, LMs encounter numerical attributes only in form of largely unordered and unstructured textual mentions. This raises the question if LMs learn to represent numerical attributes appropriately, i.e., according to their natural structure.

Here, we devise a simple method for identifying and manipulating linear representations of numeric attributes in LMs. We find low-dimensional subspaces that strongly correlate with numeric attributes across models and numeric properties, thereby confirming and extending prior observations of linear representations (Gurnee and Tegmark, 2023). Going beyond prior work, we show that by causally intervening along certain directions in these subspaces, LM output changes correspondingly. For example, an entitity's year of birth shifts according to the strength and polarity of the intervention along a "birthyear" direction (Fig. 1). Taken together, our findings suggest that LMs represent numeric attributes in a way that reflects their natural structure and that linear representations reliably 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 §4.

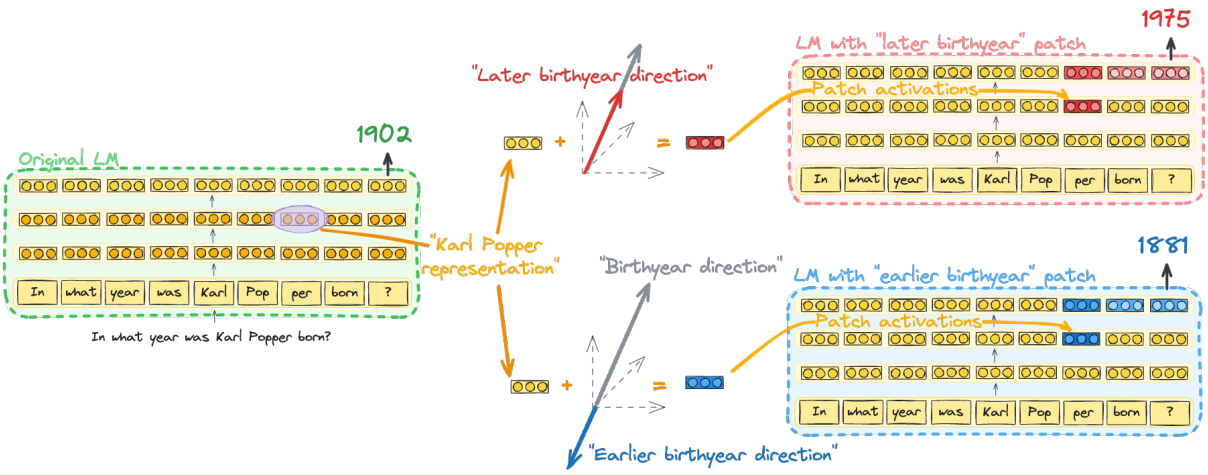


Figure 1: Sketch of our main finding. Patching entity representations along specific directions activation space yields corresponding changes in model output.

Terminology. Before moving to the main part, we briefly clarify important terms. A **quantity** consists of scalar numeric value paired with a **unit** of measurement. The value is called **magnitude**. A **numeric property** is a property that can naturally 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`.

2 Finding Attribute-encoding Directions

Motivation. While numeric attributes generally can be mapped naturally onto simple canonical structures, such as number lines or coordinate systems, it is not immediately obvious that pretraining on largely unstructured data enables LMs to appropriately represent such structures. Our main goal is to find out if and how numeric attributes, such as an entity's birthyear, are encoded in the geometry of LM representations. How could such an encoding look like? Based on two arguments, we hypothesize that numeric attributes are encoded in low-dimensional linear subspaces of an LM's representation 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 attributes (Petroni et al., 2019), we can expect that their representations reflect the structure of numeric attributes. And since the natural structures of many numerical attributes are low-dimensional linear spaces such as number lines, we expect to find low-dimensional linear 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 of representation space (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 attributes are encoded in linear subspaces. For brevity, we will call a low-dimensional linear subspace of an LM's representation space a *direction*, regardless of whether it is one- or multi-dimensional.

Method. Motivated by the hypothesis that numerical attributes are encoded as directions in representation space, we now devise a method for finding out if such directions exist. An obvious choice for identifying linear structure would be principal component analysis (PCA; Pearson, 1901). However, PCA looks for directions of maximum variance, while we are interested in directions that are predictive of model outputs. Put differently, PCA is unsupervised since we cannot provide model outputs to steer the algorithm towards directions that covary with this signal. Instead, we would like to find directions in a supervised fashion, i.e., using model outputs as supervision signal. A standard method learning the relationship between two spaces X and Y is partial least squares regression

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

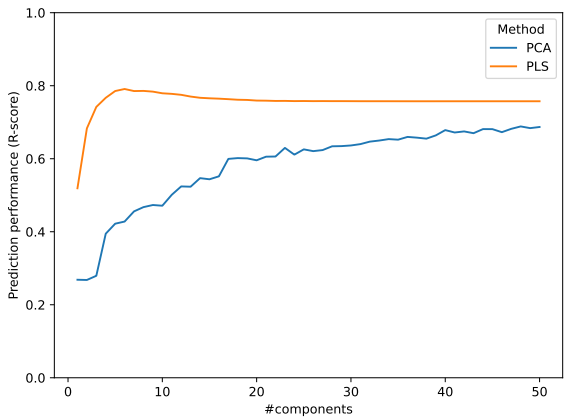


Figure 2: A low-dimensional subspace of Llama-2-13b's 5120-dimensional activation space (in layer $l = 0.3 = 12/40$) is highly predictive of the quantity expressed by the LM (output: "1902") when queried for the birthyear of entities (prompt: "When was Karl Popper born? One-word answer only:"). Unlike regression on PCA components (blue), partial least squares regression (PLS, orange) identifies a small set of predictive components.

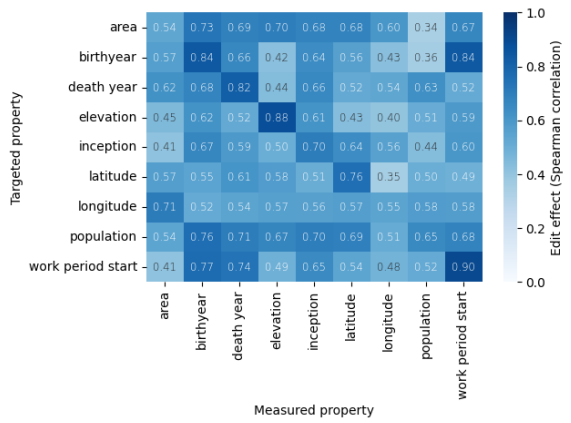


Figure 3: Mean-aggregated effects and side effects when performning activation patching along property-specific directions in activation space. Diagonal entries (top-left to bottom right) show the effect on the targeted property in terms of mean Spearman correlation between edit weight w_i, k and expressed quantity y_i, k. For example, patching an entity representation along a "birthyear" direction results in a corresponding change in the Llama-2-13b's expressed quantity with a correlation strength of 0.84. Off-diagonal entries show the side-effects of activation patching, e.g., "birthyear" patches affect LM output when queried for an entity's death year with a correlation strength of 0.68.

(PLS; Wold et al., 2001). In our case X corresponds to the representation space of an LM and Y to LM outputs. Concretely, for a given numeric attribute, e.g., birthyear, we collect n entity representations of dimension d, i.e., $X \in R^{nxd}$, and the quantity expressed by the LM when queried for the numeric attribute of each entity, i.e., $Y \in R^n$. We then fit a PLS model, varying the number of components, i.e., the dimensionality of the subspace, and record the prediction performance of the resulting model in terms of R-score.

Results. We randomly sample $n = 1000$ popular[3] person entities from Wikidata (Vrandečić and Krötzsch, 2014), prompt the LM (in English) for their birthyear and take the d-dimensional hidden state of the entity mention's last subword token[4] at layer l to obtain a sample of the representation space X, choosing l on a development set. We record the n quantities parsed from the LM output to obtain Y and fit PLS models with varying numbers of components to find birthyear-encoding directions and show results for Llama 2 7b in Fig. 2. We find that birthyear output can be predicted from a small number of PLS components with the highest R-score (0.79) obtained with a 7-dimensional direction in Llama 2's representation space.

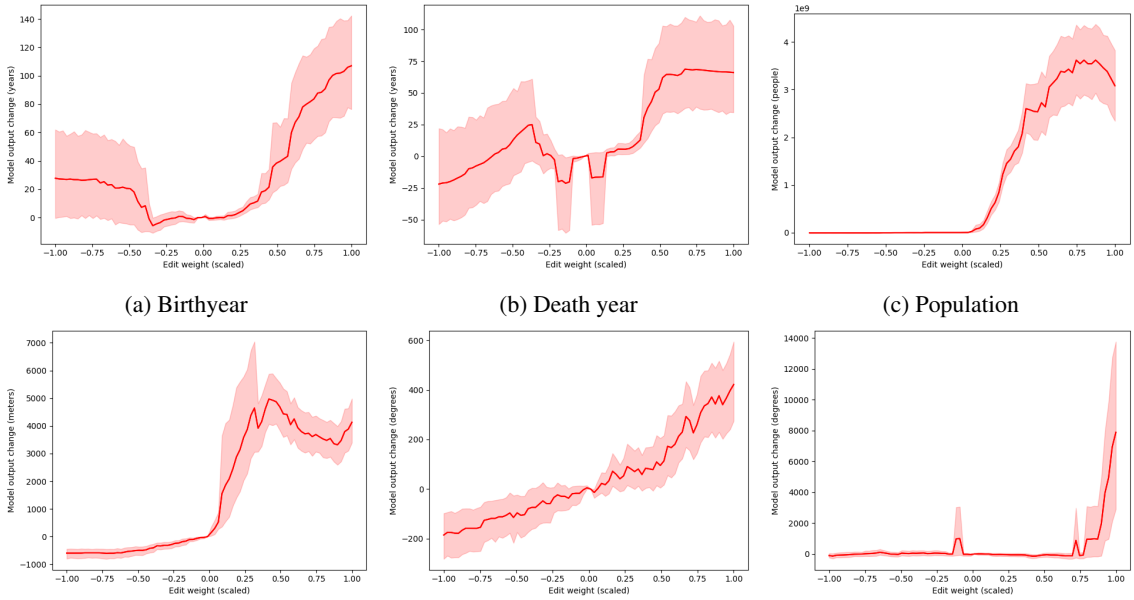
[3]We define popular entities as those in the top decile in terms of Wikidata node degree and Wikipedia article length.

[4]Following Geva et al. (2023).

3 Patching Attribute-encoding Directions

Motivation. So far, we have found correlative evidence for the existence of directions in representation space that encode numeric attributes. Concretely, partial least squares regression on entity representations found directions that are predictive of what quantity the LM will express when queried for a particular numeric attribute of a particular entity. However, representation is not a sufficient criterion for computation (Lasri et al., 2022). In our case this means that numeric attributes might be encoded in representations without affecting model output. In order to make the stronger claim that numerical attributes are not only represented linearly, but that these linear representations affect model output, we now perform interventions to establish causality.

Method. Viewing the LM computation graph as causal graph (Meng et al., 2022; McGrath et al., 2023), we intervene on model activations and observe the effect on model output. Specifically, for each of the top K directions $\vec{v_k} \in R^d, k \in [1, ..., K]$ found by PLS as described in the previous section, we prepare patches $\vec{p}_{w_i,k} = w_i \vec{v_k}$ with weights w_i and edit step index $i \in [1, ..., S]$, follow-



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

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

Figure 4: Effect of activation patching along property-specific directions across several numeric properties.

ing (Matsumoto et al., 2022). While the scale of the edit weights w_i is arbitrary in principle, we empirically set their range according to the minimum and maximum PLS loadings and choose $S = 80$ edits steps for computational reasons. After sampling $n = 1000$ popular entities for nine frequent numeric properties[5] in Wikidata we first fit PLS models for each property, then apply activation patches $\vec{p}_{w_i,k}$ to the last-mention-token representations of 100 held-out entities in relative layer $l = 0.3$ and finally record the LM's expressed quantity $y_{w_i,k}$. The held-out entities are entities which were not used to fit PLS models. To operationalize the intuitions that small edit weights w_i, k should have a small effects, that negative weights should decrease and positive weights should increase the expressed quantity $y_{w_i,k}$, we quantify the intervention effect via the ranked Spearman correlation $\rho(w_{i,k}, y_{w_i,k})$.

Results. We are interested in the effects and side effects on model output when patching model activations along property-specific directions. The aggregated effect and side-effect sizes of our causal interventions are shown in Fig. 3. We observe that there are properties for which directed activation patching is highly effective, e.g., birthyear (0.84), elevation (0.88), or work period start (0.90), suggesting that these properties have highly linear representations. Other properties show a much smaller degree of linear editability, e.g., longitude (0.55) and population (0.65), suggesting that LM representations do not encode these properties well. Side effects shown on the off-diagonal entries of Fig. 3. While effects, with an average size of 0.85 ± 0.07, are generally much larger than side effects(average side effect size 0.58 ± 0.18), it is also evident that our directed activation patches can have large effects on other, non-targeted attributes.

Individual edit curves for Llama 2 13b, showing the relationship between edit weights $w_{i,k}$ (scaled to unit range) and resulting changes in the expressed quantity $y_{w_i,k}$, are shown in Fig. 4 and with curves for additional models in Appendix A.

4 Conclusion

We performed interventions on model activations to find evidence that LMs learn linear representations of numeric attributes. The representations are linear not in the sense that there is a linear relationship between intervention strength and model outputs, but in the sense of linear subspaces–in the simplest case one-dimensional directions–of activation space that stand in a causal relationship with the quantities expressed by an LM when prompted for numeric attributes.

[5]The properties are shown in Fig. 3. Longitude and latitude are non-linear, but we include these properties since they (1) are locally approximately linear almost everywhere and (2) have high coverage in Wikidata.

Limitations

While we found clear evidence for linear representation of numeric attributes, it is also clear that our simple method of activation patching along one-dimensional directions is too simple, since PLS found that numeric attributes are best predicted from low- but not one-dimensional subspaces. It is thus likely that a more refined editing method will allow more precise control over LM output. Furthermore, our analysis is limited to popular entities and frequent numeric properties, which are most likely to be "known" and well-represented by LMs. It is likely that our findings do not apply to less popular entities.

Ethics Statement

We do not foresee any ethical issues arising from this work.

References

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.

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.

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.

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.

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

Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. 2023. 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.

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

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.

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.

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.

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.

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.

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

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.

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.

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.

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.

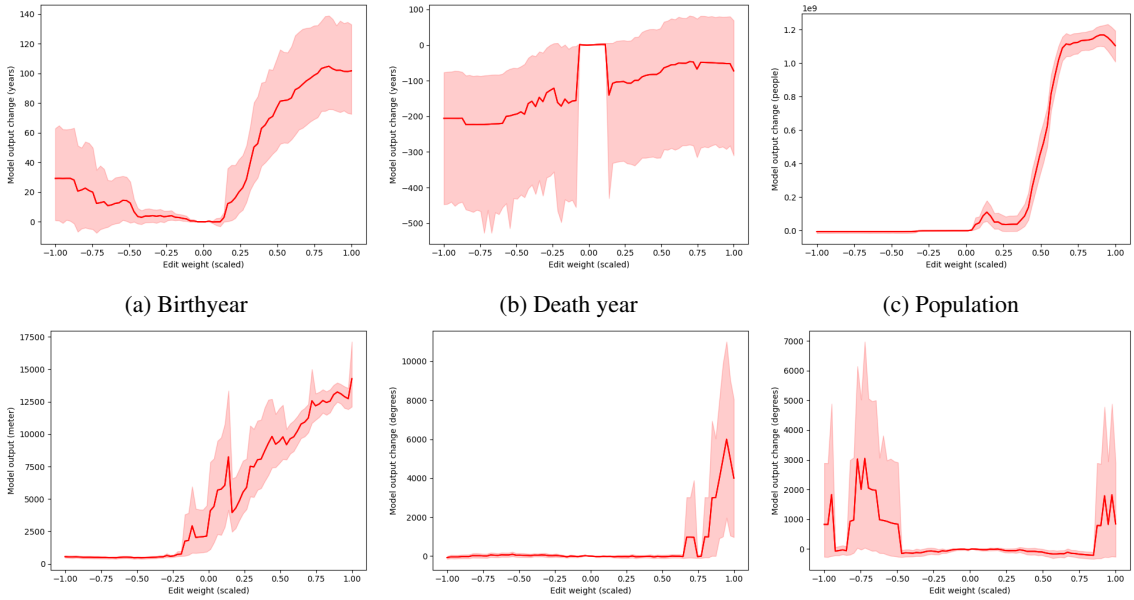
Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, 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.

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

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

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.

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.

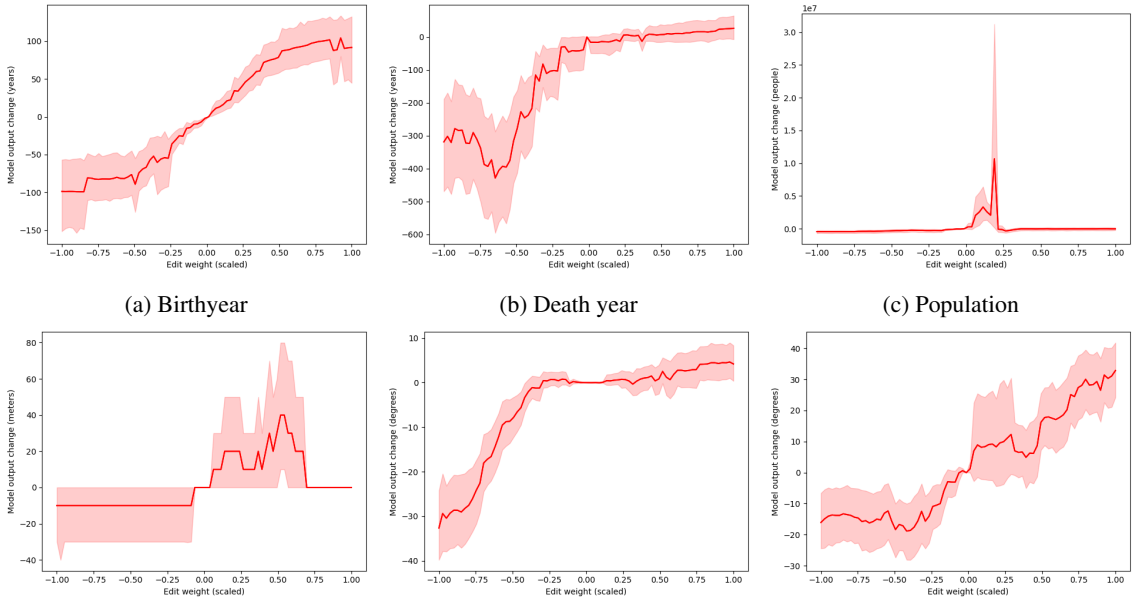


(a) Birthyear (b) Death year (c) Population

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

Figure 5: Effect of activation patching along property-specific directions across several numeric properties with LLama 2 7b.

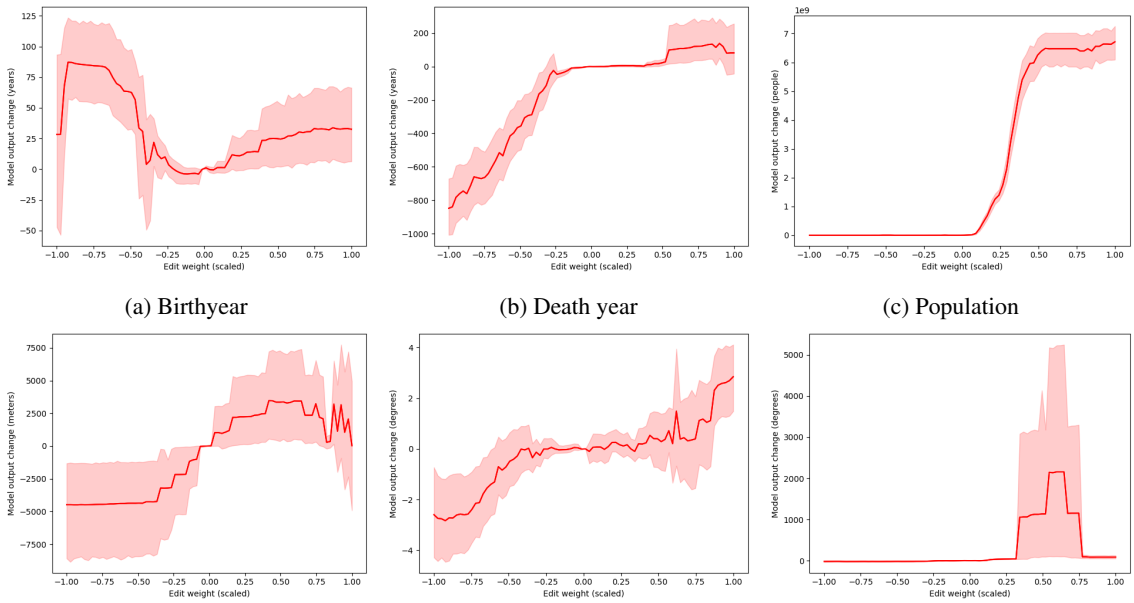
A Edit curves for additional language models



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

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

Figure 6: Effect of activation patching along property-specific directions across several numeric properties with Falcon 7b.



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

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

Figure 7: Effect of activation patching along property-specific directions across several numeric properties with Mistral 7b.