What Helped with Huge Data Hurt with Small Data: Representational Geometry Is Different for Models Trained on Cognitively Plausible Amounts of Data

Anonymous ACL submission

Abstract

Training models on cognitively plausible (CP) amounts of data (tens or hundreds of millions of words) offers multiple advantages over the huge-data training (hundreds of billions to trillions of tokens) typical of current models, but achieving competitive performance at CP scale remains an open challenge. Understanding differences in representational geometry between CP- and huge-data models may help us design better CP-scale models.

This paper studies standardization, a technique to address anisotropy and rogue dimensions in LLM representational geometry. Although standardization improves lexical similarity performance in huge-data models, the effect reverses for CP-data models. We also find that both rogue dimensionality and anisotropy correlate with data scale, further supporting the view that CP- and huge-scale models differ geometrically. Interestingly, however, statistical analyses show that anisotropy mediates only 20% of the observed difference in standardization effect, and rogue dimensionality did not significantly mediate at all. Thus, the observed qualitative difference in standardization effect hints at as-yet undiscovered differences in the geometry of CP-scale models.[1]

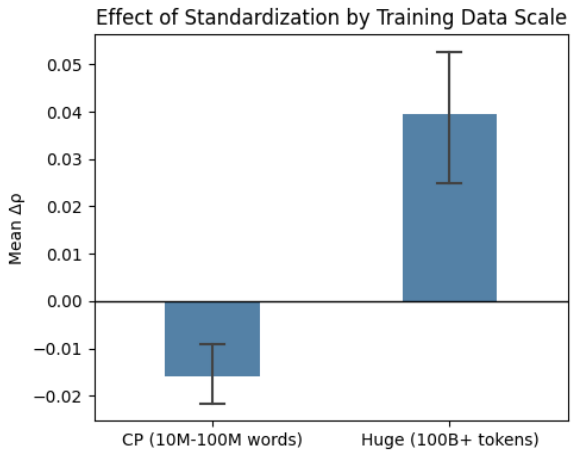

Figure 1: Standardization, a technique that helps models trained on huge data perform better at lexical similarity tasks, hurts performance for models trained on cognitively plausible (CP) data scales.

1 Introduction

While current large language models are trained on hundreds of billions or even trillions of tokens, humans learn language from a few tens or hundreds of millions of words. Training models on such *cognitively plausible* (CP) data scales offers multiple benefits compared with training on typical huge data scales. First, it may help us understand what is necessary to learn language, leading to a better alignment between NLP and cognitive science. Second, it can democratize AI: reducing the FLOPs required to train allows more people outside of industry giants to experiment. Third, it offers a viable path for NLP in less-resourced languages that do not have hundreds of billions of tokens of text available for training; it likewise benefits those who train models in well-resourced languages while respecting the intellectual property of authors.

Understanding the representational geometry of models trained on CP data can help us make progress on improving these models. Lexical similarity is a good way to see what's happening in the representational geometry, because it depends so much on the cosine similarity of vectors and evaluates them against human judgments. It's also a well-studied task with quality, human-generated benchmarks for evaluation.

With the advent of contextual embeddings and models trained on huge datasets, a number of techniques for improving lexical semantics results have emerged, with a great deal of attention paid to isotropy (uniform distribution of words within the embedding space) and its opposite, anisotropy

[1] Data and code available at [*redacted for blind review*].

(word vectors clustered in the same direction) (Bommasani et al., 2020; Vulić et al., 2020; Mahajan et al., 2025). (Timkey and van Schijndel, 2021) observed that anisotropy was often attributable to just a few *rogue dimensions*; they addressed the problem with vector standardization, which improved performance on lexical similarity for several large models.

We find, however, that **despite being effective for models trained on huge data, standardization actually *harms* lexical similarity results for models trained on cognitively plausible data**. Given the motivation for vector standardization in work with huge data models, we hypothesize that rogue dimensionality or anisotropy may explain this. Surprisingly, though, while both geometric phenomena are correlated with training data scale, our statistical analyses show that rogue dimensionality does not explain the reversal in effect and that anisotropy mediates only 20% of the effect. This suggests that the geometry of models trained on CP-scale data differs from those trained on huge-scale data in ways not captured by existing measures.

Our contributions are as follows:

- Empirical results showing that standardization improves lexical similarity performance on models trained on huge data but harms it for models trained on CP data.
- Results showing that rogue dimensionality and anisotropy are correlated with how much data the model was trained on.
- Analysis showing that rogue dimensionality does not explain the reversal of standardization's effect on the CP-data vs. huge-data models, and anisotropy explains only 20%.

In short, we have discovered a qualitative difference in the geometry of embedding space between CP- and huge-data models, revealing a gap in our field's understanding of how CP-data models learn.

2 Background & Related Work

2.1 Lexical Similarity and Representational Geometry

Lexical similarity is the task of determining how similar two words are. Distributional word embeddings in the 2010s showed great improvement over previous methods (e.g., (Mikolov et al., 2013)). Getting further gains from LLMs has proven challenging, as naive approach of passing a single word through an LLM and selecting a layer as its representation is ineffective (Bommasani et al., 2020; Ranjan, 2025; Vulić et al., 2020).

Some of this difficulty arises from the anisotropy of the representational space of LLMs. In an isotropic space, word representations are distributed uniformly in all directions from the origin; as a result, randomly selected pairs of words are not terribly similar. However, in an anisotropic space, representations cluster in a cone. (Ethayarajh, 2019) showed high anisotropy in BERT, ELMo and GPT-2; (Mahajan et al., 2025) found similar results for more recent LLMs.

(Timkey and van Schijndel, 2021) found that a small subset of 1-5 "rogue dimensions" drive anisotropy. They found that the top three dimensions contributed up to 98% of cosine similarity between random word pairs; likewise, rogue dimensions drove the variability in cosine similarity. They found that a simple post-processing vector standardization—subtracting the mean and dividing by the standard deviation—consistently improved performance of several LLMs on multiple lexical similarity tasks.

Other work notes that anisotropy is beneficial for some purposes, and increasing isotropy is not always desirable. (Raunak et al., 2020) note that although post-processing to increase isotropy had improved results for semantic similarity tasks, it harmed performance for sentence classification and machine translation tasks. (Rudman and Eickhoff, 2024) proposed an alternative measure of isotropy and found that increasing isotropy according to this measure led to decreased performance on several common NLP benchmarks.

2.2 Data Scales

The scale of data used to train modern LLMs is enormous and growing; for example, Llama 3 was trained on 15 trillion tokens (Grattafiori et al., 2024), while Llama 4 trained on 30 trillion tokens (AI, 2025). Even more open models, such as Pythia (Biderman et al., 2023) and Gemstones (McLeish et al., 2025) train on hundreds of billions of tokens, and Olmo 3 base trained on 6 trillion tokens (Olmo et al., 2026).

In response to this, some in the NLP community have proposed studying how a computational system can learn language from data at a *cognitively plausible* scale (Choshen et al., 2026; Charpentier et al., 2025; Hu et al., 2024; Warstadt et al., 2023)—that is, at a scale comparable to what a human child

hears while learning language. Models trained at smaller scale offer many advantages: They may offer better insights into how the human brain learns language. They can be trained more quickly and on less expensive hardware than models trained on huge data. They could enable more functional NLP in less-resourced languages. The annual BabyLM challenge operationalizes cognitively plausible data scales as *strict* (100M space-separated words) and *strict-small* (10M words), roughly corresponding to the number of words a child of 12 years old (strict) or 2-5 years old (strict-small) has been exposed to in his or her lifetime (Warstadt and Association for Computational Linguistics, 2023).

Relatively little work has considered whether models trained on smaller scale data differ from those trained on huge scale data in kind rather than in degree. Previous work on qualitative differences at scale refer to model scale, not data scale (e.g., (Wei et al., 2022)). (Xiao, 2025) identified the phenomenon of *scaling law crossover*, where the techniques that benefited small models harmed larger models. While motivated by differences in training needs between small and large *data* scales, the experiments conflate data scale and model scale, preventing conclusions about which explains the crossover.

A few recent works have examined how LLM characteristics vary over training checkpoints. (Machina and Mercer, 2024) looked at trends in isotropy over 21 evenly spaced checkpoints of several Pythia models. (Li et al., 2026) identified phases that the geometry of representational space passed through during training of large (160M-12B parameter) models. (Müller-Eberstein et al., 2023) studied when during training various types of linguistic information was encoded in their models.

These aim to understand how and when desirable properties of models trained on large data emerge. To the best of our knowledge, no one has explored the opposite perspective: what might qualitative differences between models from small and large data regimes tell us about how best to train on cognitively plausible data scales?

3 Methods

3.1 Models

We use two main types of models for our experiments: those trained only on CP (strict- or strict-small) datasets, and models that release a sequence of checkpoints that have seen small through huge amounts of training data. For the CP-data-only models, we have a custom baseline, as well as models released as baselines for the BabyLM challenge in 2023, 2024, and 2025. For checkpoint sequences, we use Pythia and Gemstones. In selecting the checkpoint-sequence models, we focus on small versions (tens of millions of parameters), as Chinchilla-style scaling (Hoffmann et al., 2022) suggests this would be the parameter count most appropriate for models to be trained on CP-scale data, and this parameter count is most directly comparable to that of the CP-data models. Model sizes and training data scale are summarized in Table 1.

The Pythia family of decoder-only models (Biderman et al., 2023) was developed to help researchers study how LLMs change over the course of their training and how scale affects those changes. Full Pythia models are trained on 300B tokens of the Pile (Gao et al., 2020). We use two models, Pythia-14M and Pythia-70M. We report the results of the fully trained version of each, as well as all checkpoints.

Gemstones (McLeish et al., 2025) is a similar suite of models, with more variations in architectural shapes and hyperparameters. Their architecture is decoder only and based on Gemma (Team et al., 2024). We use Gemstone-256x23, (256 dimensions and 23 layers) and Gemstone-512x11 (512 dimensions and 11 layers). As with Pythia, we report results across all checkpoints of Gemstone, up to the final checkpoint trained on 350B tokens of Dolma 1.7 (Soldaini et al., 2024) data.

Our custom BaselineLM is a simple causal language model using next-token prediction as its objective function. We train versions on two 10M-word datasets: a 50-50 split of text from LocalizedNarratives (Pont-Tuset et al., 2020) and the English subset of WIT: Wikipedia-based Image Text Dataset (Srinivasan et al., 2021), [2] and a subset of the Pile (Gao et al., 2020).[3] Our BaselineLM model was therefore trained on two different distributions of text: one more representative of what a multimodal model might encounter, and one from approximately the same distribution as Pythia models. In both cases, the model was trained on approx-

[2] Although these datasets offer images associated with the text, we do not use those images in training.

[3] We used the `monology/pile-uncopyrighted` version, `https://huggingface.co/datasets/monology/pile-uncopyrighted`, which excludes subsets of the Pile known to contain copyrighted materials. We used stratified sampling based on the original proportions of subsets (excluding copyrighted ones) to collect 10M words of text.

Model	Parameters	Data Scale
Pythia-14M	14M	huge
Pythia-70M	70M	huge
Gemstone 256x23	48M	huge
Gemstone 512x11	95M	huge
BaselineLM	35M	CP (strict-small)
BabyLlama	58M	CP (strict & strict-small)
OPT-125M[4]	163M	CP (strict)
GPT-BERT 100M (causal, masked, & mixed focus)	132M	CP (strict)
GPT-BERT 10M (causal, masked, & mixed focus)	39M	CP (strict-small)
GPT-2	111M	CP (strict & strict-small)

Table 1: Models used in experiments, with their parameter counts and training data scale. Following the BabyLM challenges, "strict-small" means 10M space-delineated words of training data and "strict" means 100M words; both are cognitively plausible (CP). "Huge" here refers to hundreds of billions of tokens.

imately 10M words (split on spaces) of text.

All remaining models were trained on the BabyLM train sets. BabyLlama (Timiryasov and Tastet, 2023) used knowledge distillation from an ensemble of GPT-2 and LLaMA models; BabyLlama outperforms its teachers and a comparable non-distilled model trained on the same data. Opt-125M (Zhang et al., 2022) is a decoder-only model used as a baseline in the first BabyLM challenge (Warstadt et al., 2023). GPT-BERT (Charpentier and Samuel, 2024) uses a hybrid masked language modeling and causal language modeling objective. It was the winner of two tracks for the BabyLM shared task in 2024 and was used as a baseline for the 2025 shared task. We use the models released on Huggingface: GPT-BERT causal focus (where the causal objective is weighted more heavily), masked focus (where the masked objective is weighted more heavily), and mixed (where the two objectives are evenly weighted). Finally, we use the versions of GPT-2 Small (Radford et al., 2019) trained as baselines for the BabyLM 2025 challenge (Charpentier et al., 2025).

When we refer to experiments as "across models," we mean that we included the final checkpoint of each model, fully trained on the data scales described in Table 1 for a total of 18 models. Experiments are "within models" when we use the published checkpoints of the Pythia and Gemstone models.

3.2 Lexical Similarity

We use SimLex-999 (Hill et al., 2015) for our evaluation. This well-established dataset contains 999 word pairs with similarity scores based on human annotations.

The established approach to lexical similarity of two words, A and B, is to take the cosine similarity of the vector representations for A and B. Performance is measured by Spearman's ρ correlation between the cosine similarities and the gold similarity scores; high ρ indicates better performance, and human inter-annotator agreement for SimLex was $\rho = 0.67$.

Large language models introduce some options for what vector representations to use. First, it is not obvious which layer or layers encode the relevant information; some work supports using layer 0, the word embedding layer (Wang et al., 2019; Conneau et al., 2020), while others support averaging across some or all layers (Liu et al., 2019; Vulić et al., 2020). (Bommasani et al., 2020) reported that lower layers typically performed best; however, as they pooled over an increasing number of contexts, the best-performing layer shifted higher in the pretrained models. We therefore evaluate on each layer of each model.

Since SimLex contains whole words and language models typically use subword tokenization, a second question arises as to which subword to use or how to combine multiple tokens in a single word. We follow (Bommasani et al., 2020), who found mean pooling of subword vectors to yield the best results.

Finally, since language models produce contextualized embeddings, one must choose the context for the words. Prior work has established that passing a word in isolation through the network is not as effective as averaging the representations obtained from the word in multiple sentences (Bommasani et al., 2020; Vulić et al., 2020). Since Vulić et al. observed that an average over 100 contexts yielded only marginal performance gains compared with average over 10 contexts, we use 10 contexts per word to allow more efficient experiments. Our context sentences are drawn from the Wikipedia dump[5] (Foundation, 2023). No effort was made to disambiguate word sense, as SimLex itself does not

[4] 125M in the name refers to parameter count without the LM head.

[5] https://huggingface.co/datasets/wikimedia/wikipedia Configuration: 20231101.en

specify a word sense. We confirmed with preliminary experiments (reported in Appendix A) that use of these context sentences significantly outperformed isolated word embeddings.

3.3 Operationalizing Anisotropy & Rogue Dimensionality

We follow Ethayarajh (2019)'s measure of anisotropy as the average cosine similarity of random pairs of tokens. A score of 1 indicates high *an*isotropy, while a score of 0 indicates high isotropy. We measure anisotropy for each layer of each model. We sample 500k pairs of tokens from a set of Wikipedia sentences; sentences in the training data for our BaselineLM are excluded, but there could be overlap with train sets for other models.

In measuring the effect of rogue dimensions, Timkey and van Schijndel (2021) first calculated the contribution of each dimension to cosine similarity for each pair of tokens, enabling them to see how much each dimension contributed to a high anisotropy score; those that contributed a great deal were rogue. They then calculated the proportion of variance in cosine similarity of all dimensions that was accounted for by cosine similarity of all *except* those rogue dimensions. They reported r^2, the square of the Pearson correlation between cosine similarity of the tokens with all their dimensions and the cosine similarity of the tokens with the top k dimensions removed. If removing a dimension had very little impact, r^2 will be close to 1. The more a given layer contributes to the variability in cosine similarity—that is, the more rogue it is—the closer to 0 r^2 will be when that dimension is removed. In other words, high values mean the dimensions are not very rogue. We refer to this measure as *rogue dimensionality*. Following Timkey and van Schijndel (2021), we report rogue dimensionality for $k \in \{1, 3, 5\}$.

3.4 Standardization

We also follow (Timkey and van Schijndel, 2021)'s definition of standardization. To standardize an embedding $x \in \mathbb{R}^d$, we subtract the mean vector from the embedding and dividing each dimension by its standard deviation: $z = (x - \mu)/\sigma$. See (Timkey and van Schijndel, 2021) equations 7-9 for details.

In the standardized condition, we apply this technique; in the not standardized condition, we do not. We use Spearman's ρ to measure how well each technique performs on the SimLex-999 task. We find the difference between Spearman's ρ in the two conditions:

$$\Delta\rho = \rho_{plain} - \rho_{standardized}$$

so that a positive $\Delta\rho$ indicates that standardization helped performance and negative $\Delta\rho$ means standardization harmed performance.

4 Experiments

Our experiments seek to address three research questions:

1. Is the effect of standardization different for models trained on CP data vs. models trained on huge data?
2. Do rogue dimensions explain the observed differences?
3. Does anisotropy explain the observed differences?

4.1 RQ1: Effects of Standardization on CP-Data vs. Huge-Data Models

4.1.1 Across Models

Method Details. We first compare the effects of standardization across a set of different models trained on varying amounts of data. We use the BabyLM baseline models and our custom BaselineLM to represent models trained on CP data, and we use the fully trained final checkpoints of the Pythia and Gemstone models to represent models trained on huge data.

For each model, we calculate mean $\Delta\rho$ over all layers, resulting in 14 scores for the CP models and 4 for the huge models. To determine whether standardization helps, hurts, or is neutral for each group, we use a two-tailed t-test.

Results/Analysis. Results are summarized in Figure 2, which clearly shows that huge-data models benefit from standardization, but CP-data models are harmed by it. Standardization significantly *reduced* lexical similarity performance for the CP-data models (mean $\Delta\rho = -0.0158$, $p < 0.001$), indicating a modest but consistent effect. In contrast, standardization significantly *improved* lexical similarity performance for the four huge-data models (mean $\Delta\rho = 0.0369$, $p = 0.015$), despite the small number of models in this category.

We cannot rule out, based only on these experiments, the possibility that differences between models in the huge-data and CP-data groups accounts

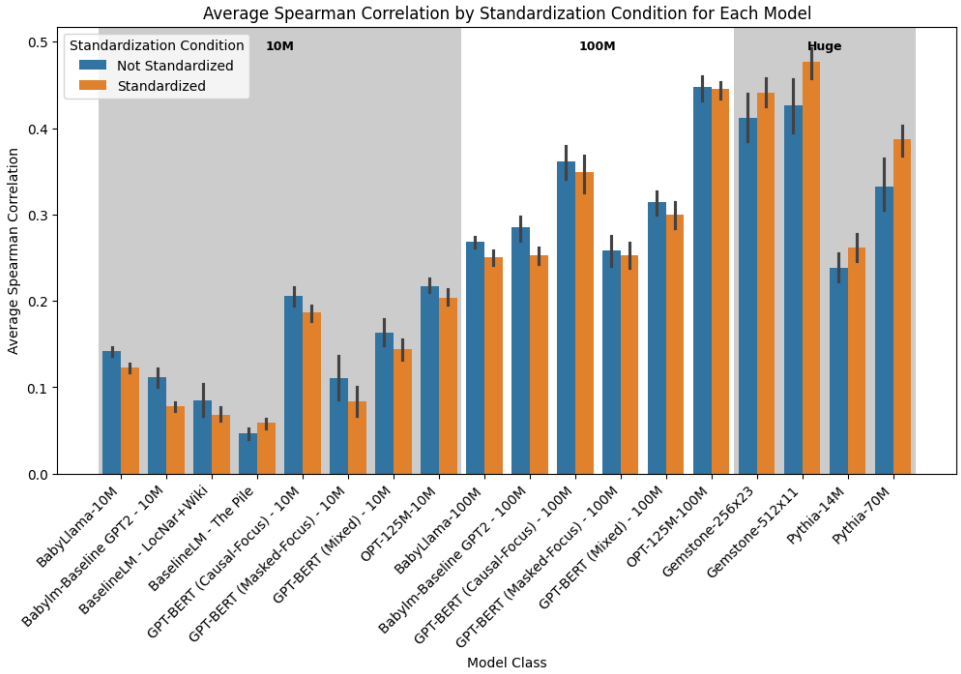

Figure 2: Models trained on 10M or 100M words performed better on a lexical similarity test without standardization (blue bars) than with it (orange bars). However, models trained on hundreds of billions of tokens benefited from standardization.

for the difference in effect of standardization. Our next experiment addresses this.

4.1.2 Within Models

Method Details. We ask whether, within the same model, the checkpoints trained on CP data are affected by standardization differently than the checkpoints trained on huge data. We evaluate each checkpoint of the Pythia and Gemstone models on SimLex-999 and calculate $\Delta\rho$ for each layer of each checkpoint. We group checkpoints trained on fewer than 10B tokens as the CP-data checkpoints and checkpoints trained on greater than 100B tokens as the huge-data checkpoints, to incorporate all scales that might be cognitively plausible and all scales that reflect how LLMs are typically trained in the other. We use a paired two-tailed t-test to determine whether mean $\Delta\rho$ for the CP-data checkpoints of each model differs significantly from the huge-data checkpoints. Given the small number of models ($n = 4$), we additionally verify using a mixed linear model regression model on all CP- and huge-data checkpoints with model name as a random effect.

Results/Analysis. In all four models, we observe that the early checkpoints are harmed by standardization, while the late checkpoints benefit from it. Figure 3 illustrates this pattern; graphs for the other models are in Appendix B. Average $\Delta\rho$ for the CP- and huge-data checkpoints is reported in Table 2. Conspicuously, the mean difference is negative for all models on CP-data checkpoints and positive for all models on huge-data checkpoints. These results are consistent with our observations on the across-model experiments. Despite the small n, the t-test indicates that the difference in average $\Delta\rho$ is significant (t=5.5980, p=0.011). The mixed-effects model findings further support this (coef=0.042, p<0.001).

Model	<10B tokens	>100B tokens
Gemstone-256x23	-0.0059	0.0232
Gemstone-512x11	-0.0002	0.0421
Pythia-14M	-0.0112	0.0219
Pythia-70M	-0.0148	0.0477

Table 2: Average $\Delta\rho$ for CP-data (<10B tokens) checkpoints was negative, while that for huge-data (>100B tokens) checkpoints was positive for all models.

We thus conclude that standardization of vectors harms performance for models trained on cognitively-plausible quantities of data and improves performance for models trained on huge data. Our next research questions investigate two possible mechanisms for this effect. Since standardization was proposed to mitigate the effects of rogue dimensions, which in turn are a form of

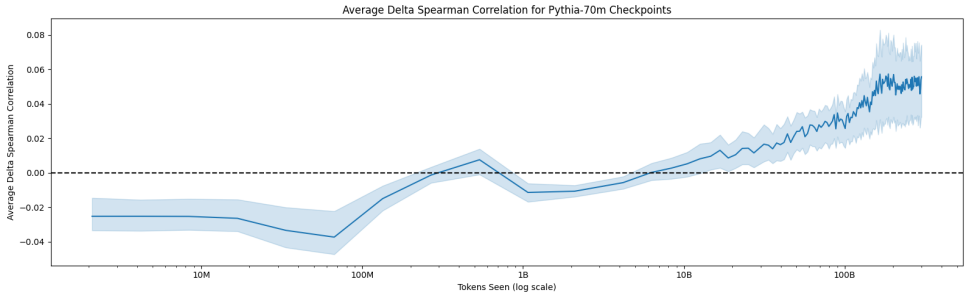

Figure 3: $\Delta\rho$, the difference between Spearman's ρ in the two standardization conditions, across number of tokens seen. Early in training, standardization harms lexical similarity performance for Pythia-70m. As the model sees more tokens, standardization becomes helpful.

anisotropy, we investigate whether differences in rogue dimensionality or anisotropy more generally account for the observed differences in standardization.

4.2 RQ2: Does rogue dimensionality explain observed differences?

Method Details. To determine whether rogue dimensionality explains the standardization difference, we first investigate whether rogue dimensionality (operationalized as described in section 3.3) is correlated with greater benefits from standardization (positive $\Delta\rho$) using Spearman's ρ across models and across layers of those models. Since results from layers of the same model may not be independently distributed, we additionally perform a mixed linear model regression with model name as a random effect. We inspect the distribution of rogue dimensionality scores for each data size.

Finally, we perform a mediation analysis (Baron and Kenny, 1986). We use a log-transformed count of tokens seen as the independent variable and $\Delta\rho$ as the dependent variable and test whether rogue dimensionality partially explains the observed effect. We use the Pingouin package (Vallat, 2018), which implements a non-parametric bootstrap method.

Results/Analysis. Across all three values of k, Spearman's ρ between rogue dimensionality and $\Delta\rho$ was negative and significant (see Table 3). This suggests that greater rogue dimensionality (i.e., lower R^2) is associated with a greater benefit from standardization, as might be inferred from the original paper. When we apply a mixed linear model with model name as a random effect, however, the correlation is not significant at $k = 1$ ($p = 0.378$). Thus, if we focus on only one rogue dimension, we cannot conclude that rogue dimensionality is correlated with standardization benefit. The mixed linear model results do agree with the Spearman's results for $k \in \{3, 5\}$, leading to the conclusion that for these k values, rogue dimensionality is correlated with standardization benefits. See Appendix C for more information on the distribution of R^2.

Metric	$k = 1$	$k = 3$	$k = 5$
Spearman	-0.293^{***}	-0.294^{***}	-0.307^{***}
MLM Coef.	-0.012	-0.042^{**}	-0.047^{***}

Table 3: Relationship between rogue dimensionality (R^2) and standardization benefit ($\Delta\rho$) across values of k. Spearman correlations computed across all models and layers; mixed linear model with model name as random effect. $^{***}p < 0.001$, $^{**}p < 0.01$.

However, mediation analysis showed no significant mediation effect for rogue dimensionality ($p = 0.992$ for $k = 1$, $p = 0.424$ for $k = 3$, $p = 0.260$ for $k = 5$). We therefore conclude that although rogue dimensionality is correlated with data size, it is *not* mediating the correlation between data size and standardization helpfulness.

4.3 RQ3: Does anisotropy explain the observed differences?

Method Details. We begin by checking if anisotropy is positively correlated with $\Delta\rho$, then perform a mediation analysis as we did for rogue dimensionality. We follow up with mediation analyses for the within-model checkpoints of the Pythia and Gemstone models.

Results/Analysis. We find a positive correlation between anisotropy and $\Delta\rho$ using both methods: the Spearman correlation is $\rho = 0.296$ ($p < 0.0001$), and the mixed linear model regression yields a coefficient of 0.084 ($p < 0.001$).

Results of the mediation analysis are summarized in Table 4. We again see a positive correlation between anisotropy and $\Delta\rho$. We observe

Path	Coef.	p
train size $\rightarrow$ anisotropy	0.0627	< 0.0001
anisotropy $\rightarrow \Delta\rho$	0.1303	< 0.0001
Total	0.0130	< 0.0001
Direct	0.0103	< 0.0001
Indirect	0.0027	0.0320

Table 4: Mediation analysis results show that train size is positively correlated with anisotropy, and anisotropy is positively correlated with $\Delta\rho$. Mediation by anisotropy accounts for approximately 20% of the total effect (0.0027/0.0130)

.

a total effect of 0.0130, with mediation accounting for 0.0027, or just over 20% of that effect. In other words, while anisotropy partially mediates the relationship of training data size and $\Delta\rho$, approximately 80% of the effect cannot be accounted for by anisotropy.

We next performed mediation analyses for results from all checkpoints of the Pythia and Gemstone models. As seen in Table 5, the strongest mediation effect was for Pythia-70M, with 0.0046 indirect out of 0.0200 total, meaning anisotropy accounts for nearly 23% of the observed effect, leaving over 75% of the correlation between train size and $\Delta\rho$ unaccounted for. In Pythia-14M, approximately 10% (0.0012 out of 0.0114) of the total effect was mediated by anisotropy. In the Gemstone models, however, train size was negatively correlated with anisotropy, leading to a negative indirect effect. Thus, in the Gemstone models, anisotropy does not account for the observed effect.

5 Discussion

From our experiments, we see a qualitative difference between models trained on cognitively plausible data scales and models trained on huge data. The effect of standardization flips its sign. One might expect this to be due to differences in rogue dimensionality or anisotropy, as those increase with data scale.[6] However, we do not see evidence that rogue dimensionality is a mediator, and anisotropy is only a partial mediator, accounting for approximately 20% of the observed effect. That leaves approximately 80% of the effect unaccounted for, suggesting that there may be other differences in the representational geometry of CP-data models that we do not yet understand.

[6]The exception is Gemstone's anomalous negative indirect effect in Table 5, which warrants further investigation in future work.

Model	Path	Coef.	p
Pythia-14M	train size $\rightarrow$ anisotropy	0.0165	0.0000
Pythia-14M	anisotropy $\rightarrow \Delta\rho$	0.0824	0.0000
Pythia-14M	Total	0.0114	0.0000
Pythia-14M	Direct	0.0102	0.0000
Pythia-14M	Indirect	0.0012	0.0000
Pythia-70M	train size $\rightarrow$ anisotropy	0.0678	0.0000
Pythia-70M	anisotropy $\rightarrow \Delta\rho$	0.0869	0.0000
Pythia-70M	Total	0.0200	0.0000
Pythia-70M	Direct	0.0155	0.0000
Pythia-70M	Indirect	0.0046	0.0000
Gem-256	train size $\rightarrow$ anisotropy	-0.0133	0.0019
Gem-256	anisotropy $\rightarrow \Delta\rho$	0.0198	0.0015
Gem-256	Total	0.0300	0.0000
Gem-256	Direct	0.0304	0.0000
Gem-256	Indirect	-0.0003	0.0400
Gem-512	train size $\rightarrow$ anisotropy	-0.0782	0.0000
Gem-512	anisotropy $\rightarrow \Delta\rho$	0.0479	0.0000
Gem-512	Total	0.0228	0.0000
Gem-512	Direct	0.0278	0.0000
Gem-512	Indirect	-0.0050	0.0000

Table 5: Mediation analysis results for within-model experiments. All models show a positive coefficient for total (that is, the path from train size to $\Delta\rho$), although in the Gemstone models, the mediation effect is negative.

Future work should explore these and other differences between models trained on cognitively plausible data and models trained on huge data, with a special focus on differences of type rather than degree. Where previous work asks what properties emerge over the course of training that make LLMs so effective, we propose that understanding the differences from the opposite side is also important: what do these differences tell us about the best ways to develop new models for cognitively plausible data scales? Rather than borrowing and adjusting the techniques of huge scale data, an understanding of qualitative differences in the two can help identify CP-data-specific techniques.

6 Conclusion

Standardization works for models trained on over 100 B words, but not for models trained on less than 10 B words. In addition to the application to this particular task (lexical similarity), it raises the possibility that techniques developed for LLMs may not generalize well to models trained on cognitively plausible data. Our results show that this is unlikely to be simply a matter of models trained on less data having lower anisotropy or rogue dimensionality. The mechanism of this effect thus remains an interesting question for future work.

Limitations

All models were transformer-based and at least partly causal language models. GPT-BERT models do additionally incorporate a masked language modeling objective, but the architectural diversity remains somewhat limited. This is, we think, a symptom of a larger lack of model diversity in the broader NLP community at present, and something we should collectively find ways to address.

We evaluated using only lexical similarity, and only one benchmark. While SimLex-999 is a well-respected, carefully designed evaluation set, results could differ using other benchmarks or other tasks.

The difference between CP (<10B tokens) and huge (>100B tokens) data for the within-models experiments is treated as a clean boundary, but these values are somewhat arbitrary. They were chosen to incorporate all scales that might be cognitively plausible in one group and all scales that reflect how LLMs are typically trained in the other and omit a middle section that does not clearly belong in either group. Moreover, the Gemstone checkpoints start at 2B tokens, much more than the 100M of the CP-scale models.

Although one goal of CP-scale models is to benefit less-resourced languages, the present work looked only at English models. This allowed us to compare CP-scale models with huge scale models, but future work should verify that the findings extend to other languages.

Ethical Considerations

Data. We used existing datasets (the Pile, Wikipedia, WIT, and Localized Narratives) and benchmarks (SimLex-999). Web-based corpora are known to have issues of author consent and bias. We attempted to reduce the former by relying on the uncopyrighted subset of the Pile.

Training. Training LLMs demands significant computation, leading to use of electricity and potential harm to the environment. We attempt to mitigate this by using primarily models pretrained by other researchers, effectively amortizing the environmental cost by getting additional scientific findings from it. The two models we trained were limited to 10M words of input and 35M parameters, making them unusually small and reducing the environmental impact of training. Moreover, we hope that studying models trained on smaller data will lead to future improvements in training efficiency, for a net benefit.

References

Meta AI. 2025. The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation. https://ai.meta.com/blog/llama-4-multimodal-intelligence/.

Reuben M Baron and David A Kenny. 1986. The Moderator-Mediator Variable Distinction in Social Psychological Research: Conceptual, Strategic, and Statistical Considerations. *Journal of personality and social psychology*, 51(6).

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling. *Preprint*, arXiv:2304.01373.

Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781, Online. Association for Computational Linguistics.

Lucas Charpentier, Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Michael Y. Hu, Jing Liu, Jaap Jumelet, Tal Linzen, Aaron Mueller, Candace Ross, Raj Sanjay Shah, Alex Warstadt, Ethan Gotlieb Wilcox, and Adina Williams. 2025. Findings of the Third BabyLM Challenge: Accelerating Language Modeling Research with Cognitively Plausible Data. In *Proceedings of the First BabyLM Workshop*, pages 399–420, Suzhou, China. Association for Computational Linguistics.

Lucas Georges Gabriel Charpentier and David Samuel. 2024. GPT or BERT: Why not both? *Preprint*, arXiv:2410.24159.

Leshem Choshen, Ryan Cotterell, Mustafa Omer Gul, Jaap Jumelet, Tal Linzen, Aaron Mueller, Suchir Salhan, Raj Sanjay Shah, Alex Warstadt, and Ethan Gotlieb Wilcox. 2026. BabyLM Turns 4 and Goes Multilingual: Call for Papers for the 2026 BabyLM Workshop. *Preprint*, arXiv:2602.20092.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Emerging Cross-lingual Structure in Pretrained Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. Association for Computational Linguistics.

Kawin Ethayarajh. 2019. How Contextual are Contextualized Word Representations? Comparing the Geometry of BERT, ELMo, and GPT-2 Embeddings. 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 55–65,

Hong Kong, China. Association for Computational Linguistics.

Wikimedia Foundation. 2023. Wikimedia/wikipedia.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *Preprint*, arXiv:2101.00027.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *Preprint*, arXiv:2407.21783.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation. *Computational Linguistics*, 41(4):665–695.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, and 3 others. 2022. An empirical analysis of compute-optimal large language model training. *Advances in Neural Information Processing Systems*, 35:30016–30030.

Michael Y. Hu, Aaron Mueller, Candace Ross, Adina Williams, Tal Linzen, Chengxu Zhuang, Ryan Cotterell, Leshem Choshen, Alex Warstadt, and Ethan Gotlieb Wilcox. 2024. Findings of the Second BabyLM Challenge: Sample-Efficient Pretraining on Developmentally Plausible Corpora. *Preprint*, arXiv:2412.05149.

Melody Li, Kumar Krishna Agrawal, Arna Ghosh, Komal Teru, Adam Santoro, Guillaume Lajoie, and Blake Richards. 2026. Tracing the Representation Geometry of Language Models from Pretraining to Post-training. *Advances in Neural Information Processing Systems*, 38:54691–54724.

Qianchu Liu, Diana McCarthy, Ivan Vulić, and Anna Korhonen. 2019. Investigating Cross-Lingual Alignment Methods for Contextualized Embeddings with Token-Level Evaluation. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 33–43, Hong Kong, China. Association for Computational Linguistics.

Anemily Machina and Robert Mercer. 2024. Anisotropy is Not Inherent to Transformers. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4892–4907, Mexico City, Mexico. Association for Computational Linguistics.

Yash Mahajan, Matthew Freestone, Naman Bansal, Sathyanarayanan N. Aakur, and Santu Karmaker. 2025. Revisiting Word Embeddings in the LLM Era. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 2686–2717, Mumbai, India. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

Sean Michael McLeish, John Kirchenbauer, David Yu Miller, Siddharth Singh, Abhinav Bhatele, Micah Goldblum, Ashwinee Panda, and Tom Goldstein. 2025. Gemstones: A Model Suite for Multi-Faceted Scaling Laws. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Max Müller-Eberstein, Rob van der Goot, Barbara Plank, and Ivan Titov. 2023. Subspace Chronicles: How Linguistic Information Emerges, Shifts and Interacts during Language Model Training. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13190–13208, Singapore. Association for Computational Linguistics.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, and 49 others. 2026. Olmo 3. *Preprint*, arXiv:2512.13961.

Jordi Pont-Tuset, Jasper Uijlings, Soravit Changpinyo, Radu Soricut, and Vittorio Ferrari. 2020. Connecting Vision and Language with Localized Narratives. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V*, pages 647–664, Berlin, Heidelberg. Springer-Verlag.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Rajeev Ranjan. 2025. One Word Is Not Enough: Simple Prompts Improve Word Embeddings. *Preprint*, arXiv:2512.06744.

Vikas Raunak, Vaibhav Kumar, Vivek Gupta, and Florian Metze. 2020. On Dimensional Linguistic Properties of the Word Embedding Space. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 156–165, Online. Association for Computational Linguistics.

William Rudman and Carsten Eickhoff. 2024. Stable Anisotropic Regularization. *International Conference on Learning Representations*, 2024:23594–23609.

Skipper Seabold and Josef Perktold. 2010. Statsmodels: Econometric and Statistical Modeling with Python. In *Python in Science Conference*, pages 92–96, Austin, Texas.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, and 17 others. 2024. Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. 2021. WIT: Wikipedia-based Image Text Dataset for Multimodal Multilingual Machine Learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2443–2449, Virtual Event Canada. ACM.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, and 89 others. 2024. Gemma: Open Models Based on Gemini Research and Technology. *Preprint*, arXiv:2403.08295.

Inar Timiryasov and Jean-Loup Tastet. 2023. Baby Llama: Knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 279–289, Singapore. Association for Computational Linguistics.

William Timkey and Marten van Schijndel. 2021. All Bark and No Bite: Rogue Dimensions in Transformer Language Models Obscure Representational Quality. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4527–4546, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Raphael Vallat. 2018. Pingouin: Statistics in Python. *Journal of Open Source Software*, 3(31):1026.

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, and 15 others. 2020. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing Pretrained Language Models for Lexical Semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics.

Yuxuan Wang, Wanxiang Che, Jiang Guo, Yijia Liu, and Ting Liu. 2019. Cross-Lingual BERT Transformation for Zero-Shot Dependency Parsing. 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 5721–5727, Hong Kong, China. Association for Computational Linguistics.

Alex Warstadt and Association for Computational Linguistics, editors. 2023. *The BabyLM Challenge at the 27th Conference on Computational Natural Language Learning - Proceedings of the BabyLM Challenge: December 6-7, 2023: CoNLL 2023*. Association for Computational Linguistics (ACL), Stroudsburg, PA.

Alex Warstadt, Aaron Mueller, Leshem Choshen, Ethan Wilcox, Chengxu Zhuang, Juan Ciro, Rafael Mosquera, Bhargavi Paranjabe, Adina Williams, Tal Linzen, and Ryan Cotterell. 2023. Findings of the BabyLM Challenge: Sample-Efficient Pretraining on Developmentally Plausible Corpora. In *Proceedings of the BabyLM Challenge at the 27th Conference on Computational Natural Language Learning*, pages 1–34, Singapore. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*.

Lechao Xiao. 2025. Rethinking Conventional Wisdom in Machine Learning: From Generalization to Scaling. *Preprint*, arXiv:2409.15156.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open Pre-trained Transformer Language Models. *Preprint*, arXiv:2205.01068.

A Isolated vs. Contextualized Word Vectors

In preliminary experiments, we compared performance of Opt-125m, BabyLlama, Pythia-14m, Pythia-70m, and the baseline LM in the isolated

and contextualized conditions. In the isolated condition, each word was passed through the model with no context. In the contextualized condition, ten sentences from Wikipedia containing the word were passed through the model; the word was represented using the average of its embeddings in those ten contexts. Averaged over all models and all layers, contextualized embeddings significantly outperformed isolated embeddings on SimLex-999 (mean Spearman's ρ 0.231 vs. -0.005; paired t-test t=18.61, p<0.001). We therefore used contextualized vectors for all subsequent experiments.

B Difference in Spearman's ρ Across Checkpoints

As described in Section 4.1.2 above, within each of the Pythia and Gemstone models, early checkpoints were harmed by standardization (negative $\Delta\rho$), and later checkpoints were helped (positive $\Delta\rho$). Results over all layers, all checkpoints, are shown in Figure 4.

C Distribution of Rogue Dimensionality Results

We observed that models trained on huge data were more impacted by rogue dimensionality than models trained on 100M-word data, which in turn were more impacted than the 10M-word models (Table 6). The distribution was also quite different, as illustrated in Figure 5. Models trained on strict-small data universally had almost no rogue dimensionality, as demonstrated by R^2 close to 1, while models trained on strict-scale or huge data had greater variance.

Data Scale	$k = 1$	$k = 3$	$k = 5$
10M	0.999	0.996	0.994
100M	0.802	0.738	0.717
Huge	0.551	0.390	0.374

Table 6: Mean R^2 by data scale and k. Lower R^2 indicates greater rogue dimensionality.

D Statistical Packages Used

- SciPy (Virtanen et al., 2020)
- Statsmodels (Seabold and Perktold, 2010)
- Pingouin (Vallat, 2018)

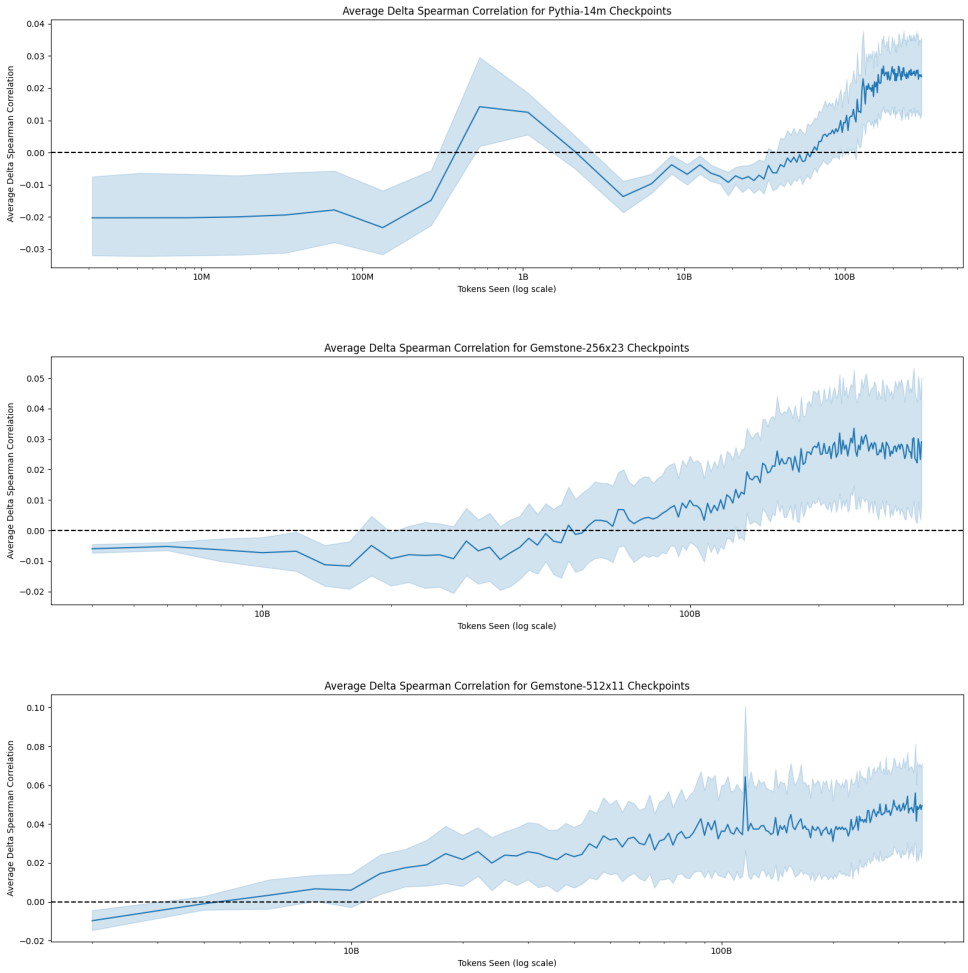

Figure 4: $\Delta\rho$, the difference between Spearman's ρ in the two standardization conditions, for Pythia-14 and the two Gemstone models. As observed in Pythia-70m (Figure 3), early in training, standardization harms lexical similarity performance. As each model sees more tokens, standardization becomes helpful.

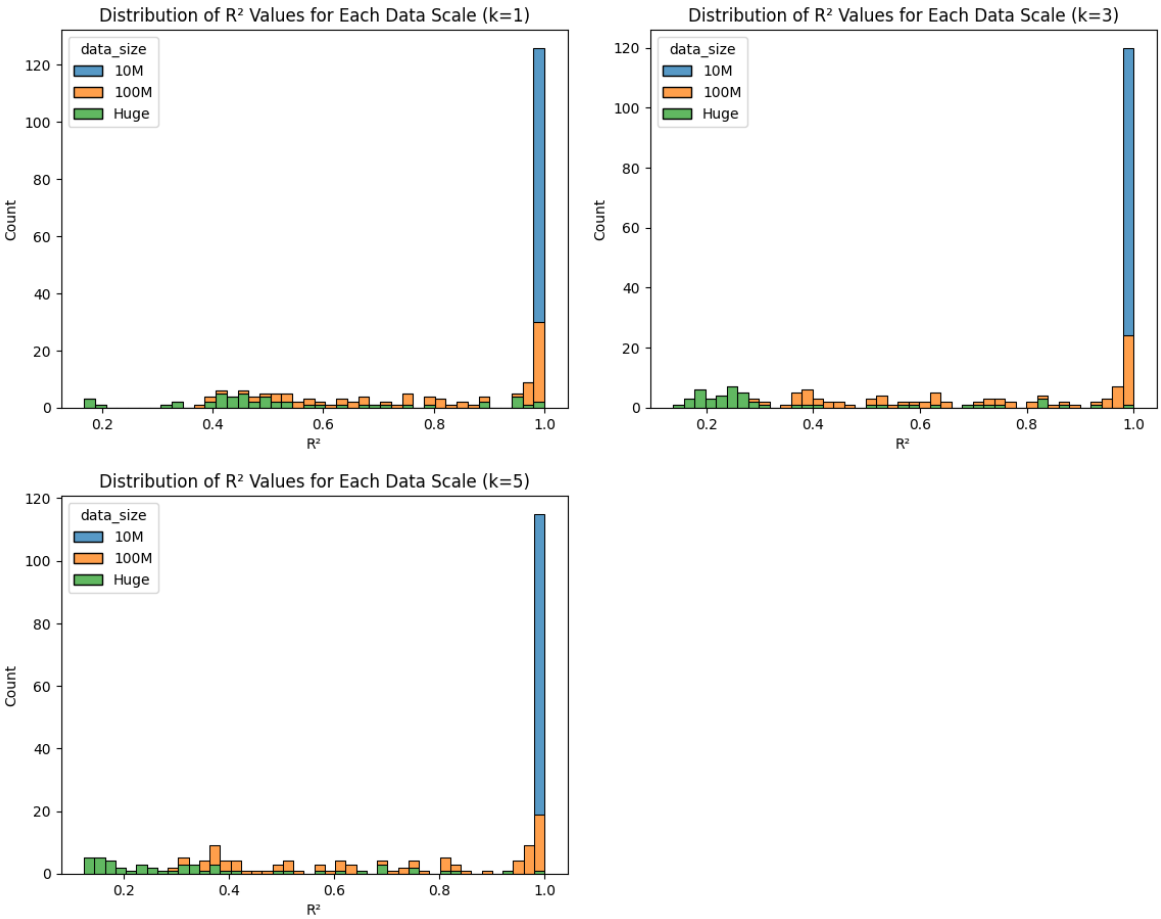

Figure 5: Distributions of rogue dimensionality results for all layers of all fully-trained models for $k \in \{1, 3, 5\}$. High R^2 indicates low rogue dimensionality. All layers of all models trained on 10M words had the lowest possible rogue dimensionality. Models trained on 100M words had less rogue dimensionality than huge models, though both showed a wider distribution.