# Anisotropy is Not Inherent to Transformers

Anonymous ACL submission


## Abstract

Isotropy is the property that embeddings are uniformly distributed around the origin. Previous work has shown that Transformer embedding spaces are anisotropic, which is called the representation degradation problem. This degradation has been assumed to be inherent to the standard language modeling tasks and to apply to all Transformer models regardless of their architecture. In this work we identify a set of Transformer models with isotropic embedding spaces, the large Pythia models. We examine the isotropy of Pythia models and explore how isotropy and anisotropy develop as a model is trained. We find that anisotropic models do not develop as previously theorized, using our own analysis show that the large Pythia models optimize their final Layer Norm for isotropy, and provide reasoning why previous theoretical justifications for anisotropy were insufficient. The identification of a set of isotropic Transformer models calls previous assumptions into question, provides a set of models to contrast existing analysis, and should lead to deeper insight into isotropy.


## 1 Introduction

Much work has found that Transformer models have globally anisotropic representations, which has been labeled the representation degradation problem (Gao et al., 2019). Isotropy has two meanings, when using cosine similarity (Ethayarajh, 2019), it means the directions of representations are uniformly distributed, and when using a partition function (Arora et al., 2016) distances must also be uniform. Anisotropy has been shown to degrade downstream task performance (Gao et al., 2019; Li et al., 2020), and an increase in isotropy correlates with better performance on some tasks. Previous work has been a set of theoretical justifications for the degradation and a large body of empirical experiments confirming global anisotropy. While no formal proof has been presented, due to the lack of any counterexamples anisotropy is often taken as assumed for any Transformer architecture.

We identify the most globally isotropic models to date, the Pythia models of size $\geq$ 410M parameters (Biderman et al., 2023), a strong counterexample to the assumption of anisotropy. These models are trained using cross-entropy loss, using autoregressive language modeling, with a final Layer Norm. Pythia model's most unique architecture feature is their untied embedding and unembeddings matrices. Pythia models have 143 evenly spaced checkpoints from training, allowing us to explore how isotropy changes during training.

We explore the isotropy of Pythia models using cosine similarity (Ethayarajh, 2019; Cai et al., 2021), a partition function (Arora et al., 2016), and our own analysis on the final Layer Norm of each model based on the theoretical work of Gao et al. (2019). Using multiple metrics allows us to present a more confident conclusion when all our isotropy measures agree. Contrary to previous work, which use token frequencies in the 1000s, we perform cosine analysis on 425M sentences from the actual training dataset, The Pile (Gao et al., 2020). This allows us to include as many rare words as possible—standard methodology ignores words with frequency less than five, and examine how isotropy might change across domains. In order to facilitate this analysis we reformulate average cosine similarity to a more computationally efficient form.

Our contributions are as follows:

- We identify a set of isotropic Transformer models: the large Pythia models.
- We analyze the isotropy of these models, both their final checkpoints and using 21 evenly spaced checkpoints during training.
- We discuss gaps in the theoretical justifications of anisotropy.

- We find that anisotropy does not happen steadily during training as previously assumed (Biś et al., 2021).

- We find that large Pythia models optimize their final layer norm for isotropy.

- We find using separate embedding and embedding weights is correlated with an increase in isotropy in large Transformer models.

## 2 Related Work

The representation degradation problem was introduced by Gao et al. (2019) for the unembedding matrix of Transformers, with a similar result discovered in a model's hidden layers (Ethayarajh, 2019) and later in sentence embeddings (Li et al., 2020). Many causes of anisotropy have been suggested, the optimal optimization solution of rare words (Gao et al., 2019), the gradient update of rare words (Biś et al., 2021), tying embedding and unembedding weights (Gao et al., 2019; Zhang et al., 2020), linguistic biases (Fuster Baggetto and Fresno, 2022), outlier neurons (Kovaleva et al., 2021; Timkey and van Schijndel, 2021), or the loss function and attention mechanisms (Godey et al., 2023b).

Most work has focused on the tied weights of the embedding (the matrix that maps tokens to input vectors) and unembedding (the matrix that maps output vectors to tokens) matrices, providing methods that increase isotropy and downstream task performance. These include token level methods focusing on the loss function (Gao et al., 2019; Wang et al., 2019, 2020a; Zhang et al., 2020), adjusting gradients (Yu et al., 2022), bias removal (Fuster Baggetto and Fresno, 2022), mean centering, PCA analysis or clustering (Arora et al., 2017; Rajaee and Pilehvar, 2022, 2021) and sentence level methods such as contrastive loss (Gao et al., 2021; Yan et al., 2021) or normalizing the mean and variance of sentence embeddings (Su et al., 2021).

Work that focuses on layers besides the unembdding layer includes cosine analysis (Ethayarajh, 2019; Cai et al., 2021), finding locally isotropic clusters (Cai et al., 2021), and "outlier neurons" found based on a dimension's contribution to cosine metrics (Timkey and van Schijndel, 2021), Layer Norm operations (Kovaleva et al., 2021), or positional embeddings (Luo et al., 2021). These "outlier neurons" can correlate with token frequency (Puccetti et al., 2022) and downstream task performance (Kovaleva et al., 2021). We note, however, that the existence of outlier neurons depends on the choice of orthonormal basis, and we could find no work linking this concept to Principal Component Analysis which should provide an orthonormal basis where the distribution of outliers correlates with the distribution of eigenvalues.

Recent work has shown that the existence of "outlier neurons" is not correlated with anisotropy (Rajaee and Pilehvar, 2022), that increases in isotropy don't necessarily correlate with downstream task performance (Ding et al., 2022), that anisotropy doesn't degrade clustering tasks (Ait-Saada and Nadif, 2023), that anisotropy causes models to rely on norm over direction (Demeter et al., 2020), and that anisotropy should only degrade results when it is caused by linguistic biases (Fuster Baggetto and Fresno, 2022).

## 3 Approach

### 3.1 Models

We use the Pythia suite (Biderman et al., 2023), a family of GPT-NeoX (Black et al., 2022) decoder only Transformer models (Vaswani et al., 2017) created by EleutherAI—comparable in architecture and number of parameters to the GPT-Neo (Black et al., 2021) and OPT (Zhang et al., 2022) models. The Pythia suite is designed with researchers in mind, providing 12 different model scales with parameters in {70M, 160M, 410M, 1.0B, 1.4B, 2.8B, 6.9B, 12B}, two models for each parameter scale—one trained on the original data and one on the deduplicated data, 144 evenly spaced training checkpoints for each model, and access to the exact dataloader used in training. We use the set of models trained on the original data, and 21 evenly spaced checkpoints from training. Pythia models use Flash Attention (Dao et al., 2022), rotary position embeddings (Su et al., 2024), parallelized attention and feed-forward (Black et al., 2022), and have separate embedding and unembedding matrices.

We also use three other models to contrast the Pythia model analysis: the OPT-6.7B model trained by Facebook (Zhang et al., 2022), which has tied embedding and unembedding matrices, Falcon-7B which uses Flash Attention and MultiQuery (Shazeer, 2019), and GPT-NeoX-20B (Black et al., 2022) which uses parallelized attention and feedforward and Flash Attention. OPT-6.7B and Falcon-

7B have tied embedding and unembeddng matrices, while GPT-NeoX-20B does not.

### 3.2 Datasets

The Pythia suite of models is trained on The Pile (Gao et al., 2020), an 825GB English language dataset originally containing 22 text sources. Recently, due to copyright claims, some text sources have been removed. To manage computation time we only use text sources that have a raw size of less than 10GB, giving us 8 different sources: Enron Emails, NIH Exporter, PhilPapers, HackerNews, EuroParl, Ubuntu IRC, DM Mathematics, and Wikipedia (en). Specific details on each source can be found in the datasheet for The Pile (Biderman et al., 2022) and in Appendix B. We use the provided dataloader to extract the sentences for each source and perform our evaluation on each text source individually and all text sources combined. We also use nine sentence classification datasets and three token level classification datasets through the SentEval Toolkit (Conneau and Kiela, 2018).

### 3.3 Layer Norm

Layer Norm (Lei Ba et al., 2016) is a common operation in transformer architectures. Given an input $\mathbf{h} \in \mathbb{R}^{\mathbf{d}}$, Layer Norm is defined as

$$LayerNorm(\mathbf{h}) = \langle \mathbf{g}, \frac{\mathbf{h} - \vec{\mathbf{1}}\mu}{\sigma} \rangle + \mathbf{b} \quad (1)$$

where $\mu$ and $\sigma$ are the mean and standard deviation of $\mathbf{h}$ and $\mathbf{g}, \mathbf{b} \in \mathbb{R}^{\mathbf{d}}$ are the trainable parameters of the Layer Norm, that is, the values of $\mathbf{h}$ are normalized with respect to mean and variance, scaled by $\mathbf{g}$, and then translated by $\mathbf{b}$. All models we evaluate ourselves have Layer Norm as the last operation before the unembedding layer.

### 3.4 Transformer Layers

While Transformer models have varying architectures (Devlin et al., 2019; Vaswani et al., 2017; Biderman et al., 2023; Brown et al., 2020) a convenient way to characterize them is as a series of layers which output a hidden state for each input token. For a given model $M$ with $L$ layers, define $H_l(s,i)$, for $l \in [0, L]$, as the function that returns the hidden state of token $w_i$ at layer $l$, where $s$ is a sentence represented as a sequence of tokens $s = \{w_1, w_2, \dots, w_n\}$. In our experiments, $H_0$ is the embedding layer, layers $H_1, \dots, H_{L-1}$ are transformer layers, and $H_L$ is the final Layer Norm operation.

### 3.5 Auto Regressive Language Models

Given a sentence represented as a sequence of tokens $s = \{w_1, w_2, \dots, w_n\}$, an auto regressive language model calculates a probability $p(s)$ by computing a product of probabilities $\prod_i P(w_t|w_{<i})$, with each term being the causal probability of a word given all previous words. The LM is then trained to maximize the log-likelihood probability

$$\max_\theta \log(p_\theta(s)) = \max_\theta \sum_{i=1}^{n} \log\left(\frac{\exp(\langle H_L(s,i), \mathbf{W_{y_i}}\rangle)}{\sum_{j=1}^{|V|}\exp(\langle H_L(s,i), \mathbf{W_j}\rangle)}\right) \quad (2)$$

where $\theta$ is the model's parameters, $V$ is the vocabulary of the model, $y_i$ is the target label for $w_i$ in $V$, $\mathbf{W} \in \mathbb{R}^{|\mathbf{V}|\mathbf{xd}}$ is the unembedding matrix, $d$ is the size of the hidden states, and $\langle ., . \rangle$ is the dot product. Note that $H_l(s,i)$ is a function of $\{w_1, \dots, w_{i-1}\}$.

### 3.6 Metrics

#### 3.6.1 Partition Functions

We use the partition function from (Arora et al., 2016) defined as

$$Z(c) = \sum_{i=1}^{|V|} \exp(\langle c, \mathbf{W_i}\rangle) \quad (3)$$

and then estimate isotropy with the function

$$I(\mathbf{W}) = \frac{\min_{\mathbf{c}\in\mathbf{X}} \mathbf{Z(c)}}{\max_{\mathbf{c}\in\mathbf{X}} \mathbf{Z(c)}} \quad (4)$$

where we use the standard approach (Mu and Viswanath, 2018; Wang et al., 2020b; Biś et al., 2021) and take $\mathbf{X}$ to be the eigenvectors of $\mathbf{W^TW}$. If $\mathbf{W}$ is isotropic then $Z(c)$ should be constant so $I(\mathbf{W})$ should be 1. In our case, $\mathbf{W}$ may be either the embedding or unembedding matrix.

#### 3.6.2 Average Cosine Similarity

Given a set of vectors $U$, where $|U| = n$, we compute the average cosine similarity between the distinct vectors, i.e.,

$$\overline{U} = \frac{1}{n^2 - n} \sum_{i=1}^{n} \sum_{j \neq i} \cos(u_i, u_j) \quad (5)$$

$$\cos(u_i, u_j) = \frac{\langle u_i, u_j \rangle}{||u_i||_2 ||u_j||_2} \quad (6)$$

where $||.||_2$ is the $L^2$ norm. Denote $\hat{u} = u/||u||_2$ i.e., the unit normalization of $u$, then Equation 5 becomes

$$\begin{aligned}\overline{U} &= \frac{1}{n^2 - n}\sum_{i=1}^{n}\sum_{j\neq i}\langle \hat{u}_i, \hat{u}_j\rangle \\ &= \frac{1}{n^2 - n}\left(-n + \sum_{i=1}^{n}\sum_{j=1}^{n}\langle \hat{u}_i, \hat{u}_j\rangle\right) \\ &= \frac{1}{n^2 - n}\left(-n + \langle\sum_{i=1}^{n}\hat{u}_i, \sum_{i=1}^{n}\hat{u}_i\rangle\right)\end{aligned} \tag{7}$$

because $\forall i\ \langle \hat{u}_i, \hat{u}_i\rangle = 1$ and because of the linearity of the inner product. Thus, we can compute $\overline{U}$ using $O(n)$ operations rather than $O(n^2)$. This allows us to compute $\overline{U}$ efficiently for large sets. We compute partial sums of 1M tokens and combine them with pair-wise summation to avoid floating point arithmetic errors. In our experiments $U$ will be the set of all hidden representations for all tokens for one layer $\{H_l(s, i), \forall s, i\}$, or the set of all hidden representation for one token $t$ for one layer $\{H_l(s, i), \forall s | w_i = t\}$. We call these *InterSim*$(l)$ and *IntraSim*$(l, t)$, respectively. These metrics are essentially the same as those seen in related works that do not focus on the embedding and unembedding matrices (Ethayarajh, 2019; Cai et al., 2021), only differing in the size of our sets and phrasing the expectation in the analytical sense.

# 4 Analysis

## 4.1 Average Cosine

### 4.1.1 Final Checkpoints

We calculate the *InterSim*$(l)$ and the average *IntraSim*$(l, t)$ for all layers of the Pythia models of size 70M, 170M, 410M, 1.4B, and 6.9B. We do this analysis using the actual data the model was trained on instead of randomly sampling a text source as is common in other analysis. While we did this analysis separately for all text sources, to measure difference in isotropy, we find no significant differences and thus only report the results on all text sources combined. Due to computation constraints, the Pythia-6.9B model is evaluated on the four smallest text sources. These results can be seen in Figures 1 and 2.

We see the 70M and 170M Pythia models have relatively low *Intra-Sim* in their middle layers followed by a sharp jump in the last transformer layer and Layer Norm. The 410M model maintains a relatively low *Intra-Sim* in most of its layers with a gradual increase and then decrease near the latter layers. The 1.4B and 6.9B models, contrastingly, have high *Inter-Sim*, quite high in the case of 6.9B, in the middle layers followed by a sharp drop in the last transformer layer and Layer Norm. We see a similar trend with Average *Intra-Sim*.

### 4.1.2 During Training

As with previous analysis, we track the *Inter-Sim* (Figure 3) and average *Intra-Sim* (Figure 4) over the course of training for the Pythia models of size 70M and 410M. As we saw no significant variance in the final results across text sources, we do this analysis using the Enron Emails text source.

We see that during the middle third of training the *Inter-Sim* of the 70M model rises sharply and then continues to gradually increase for the rest of training. The 410M model instead decreases consistently for the first two thirds of training, followed by an increase and then another gradual decrease.

## 4.2 Partition Function

### 4.2.1 Model Comparisons

We follow previous work (Mu and Viswanath, 2018; Wang et al., 2020b; Biś et al., 2021) and use the function $I(\mathbf{W})$ to estimate the isotropy of the embedding and unembedding matrices of all Pythia models, and the unembedding matrix of OPT-6.7B and Falcon-7. Following Biś et al. (2021), we also calculate $I(\hat{\mathbf{W}})$, where $\hat{\mathbf{W}}$ is the matrix where the embeddings are mean-centered, to determine if our embeddings are a translated isotropic ball, as opposed to, for example, a cone. These estimates can be found in Figures 5 and 6, respectively.

The embedding layers for all Pythia models are nearly isotropic, while for model sizes $>= 410M$ the unembedding matrices, while less isotropic than the embedding matrices, are significantly more isotropic than any other model. The largest estimate from previous work is $0.52$ while Pythia's worst estimate is $0.73$ and best is $0.82$. Further, mean centering Pythia model's embeddings always improves isotropy: significantly for Pythia-70M and Pythia-170M unembedding matrices, and to near perfect isotropy for all other Pythia models, showing that they are isotropic save for a common translation as previous work has suggested (Arora et al., 2017; Rajaee and Pilehvar, 2022, 2021). Comparing against previous work and our three other models, we see GPT-NeoX has the next best isotropy estimates, but surprisingly, due to its simi-

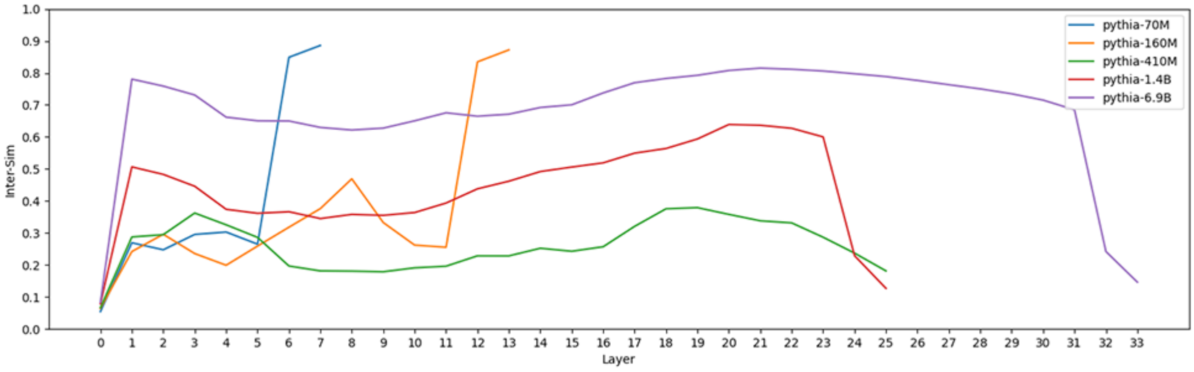


Figure 1: The *Inter-Sim*, i.e., the average cosine similarity, for each layer of the Pythia models.

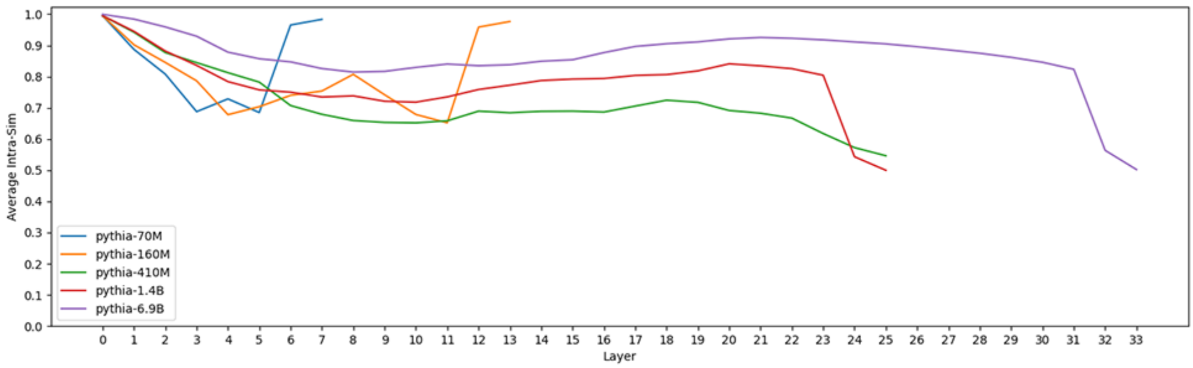


Figure 2: The average Intra-Sim over all tokens for each layer of the Pythia models.

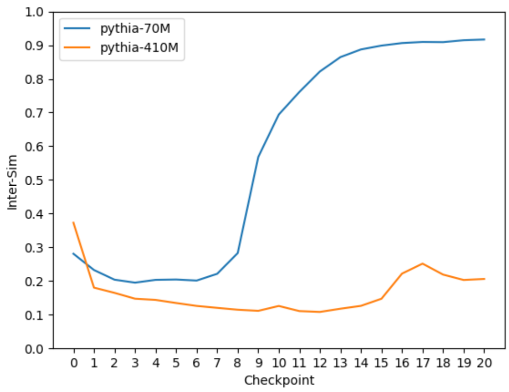


Figure 3: The *Inter-Sim*, i.e., the average cosine similarity, for the last layer of the Pythia models during training.

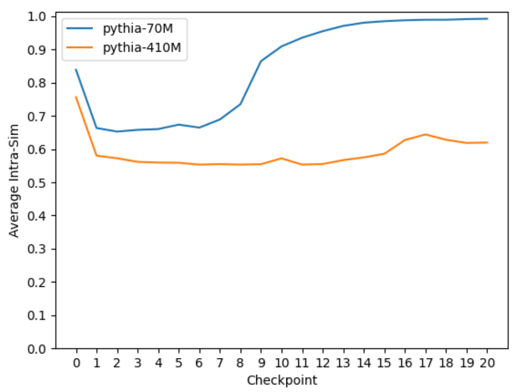


Figure 4: The *Inter-Sim*, i.e., the average cosine similarity, of all token the last layer of the Pythia models during training.

lar architecture and training, is clearly worse than large Pythia models. Falcon-7B also stands out, as mean centering did not significantly improve its estimated isotropy as it does for other auto-regressive models.

#### 4.2.2 During Training

We repeat the above analysis on the 21 evenly spaced checkpoints for the Pythia-70M, Pythia-410M, and Pythia-6.9B models. We chose these models based on the behaviours seen in the *Inter-Sim* analysis. These results can be seen in Figure 7. As the estimate for mean centering for all

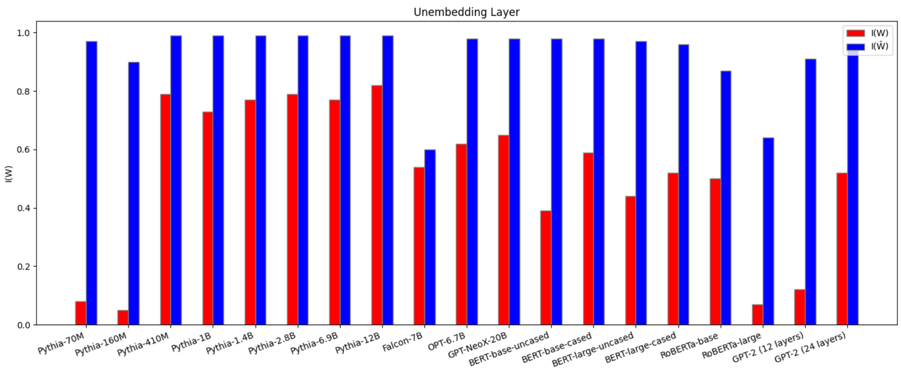


Figure 5: The I(W) calculation for the unembedding matrix $\mathbf{W}$ and mean-centered unembedding matrix $\hat{\mathbf{W}}$. BERT, RoBERTa, and GPT results are from Biś et al. (2021)

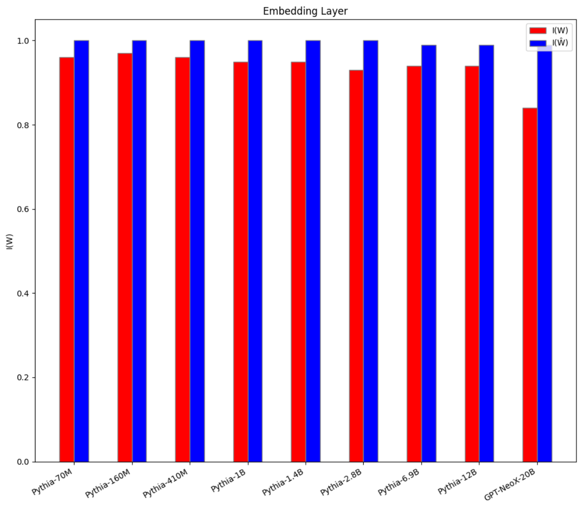


Figure 6: The I(W) calculation for the embedding matrix $\mathbf{W}$ and mean-centered embedding matrix $\hat{\mathbf{W}}$

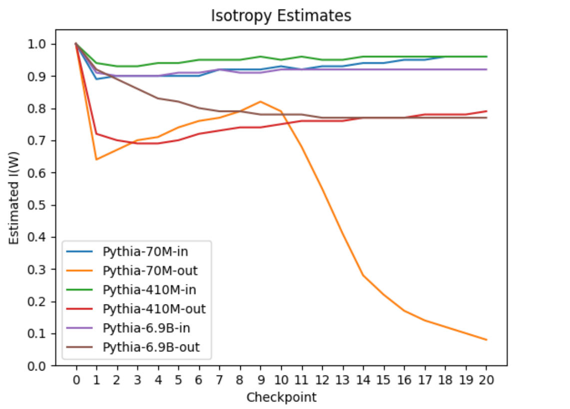


Figure 7: Isotropy estimates across 21 evenly spaced checkpoints from training, generated with the $I()$ function seen in Equation 4.

checkpoints is always nearly perfect isotropy, those results are omitted.

For the 70M and 410M models, we see a sharp drop in isotropy from the randomly initialized untrained model, and then a gradual rise in isotropy as training continues. About a third of the way into training, the Pythia-70M model's unembedding matrix continually gets less isotropic until it is almost completely anisotropic. The 6.9B model on the other hand gradually decreases and seems to stabilize around 0.77.

### 4.3 The Final Layer Norm

Due to the importance of Layer Norm in the isotropy of the hidden states of the final Layer of many transformer models (Gao et al., 2019), we analyze the parameters $\mathbf{g}$ and $\mathbf{b}$. Similar to previous works, we also analyze these parameters across training for the Pythia models of size 70M, 410M, and 6.9B.

In Figure 8 we see the average norm for the parameters $\mathbf{b}$ and $\mathbf{g}$ from Equation 1. Note that average in this case means

$$avgnorm(\mathbf{v}) = \frac{||\mathbf{v}||_2}{\sqrt{\mathbf{d}}} \tag{8}$$

as then $||avgnorm(\mathbf{v})\overrightarrow{\mathbf{1}}||_2 = ||\mathbf{v}||_2$. We see that the isotopic Pythia models have $\mathbf{b}$ parameters with the smallest norm and have the smallest ratios $||\mathbf{b}||_2/||\mathbf{g}||_2$. Figure 9 shows how the $\mathbf{b}$ and $\mathbf{g}$ parameters change during training for the Pythia models of size 70M, 410M, and 6.9B. We see a correlation between an increase in the norms of both $\mathbf{b}$ and $\mathbf{g}$ and the decrease in isotropy of Pythia-70M, whereas for the isotropic models, the norm of $\mathbf{b}$ stays low while the norm of $\mathbf{g}$ steadily increases.

We also consider the "outlier dimensions" of the Layer Norm as defined by (Kovaleva et al., 2021), however we find no correlation between the existence or not of "outlier dimensions" and isotropy, similar to Rajaee and Pilehvar (2022).

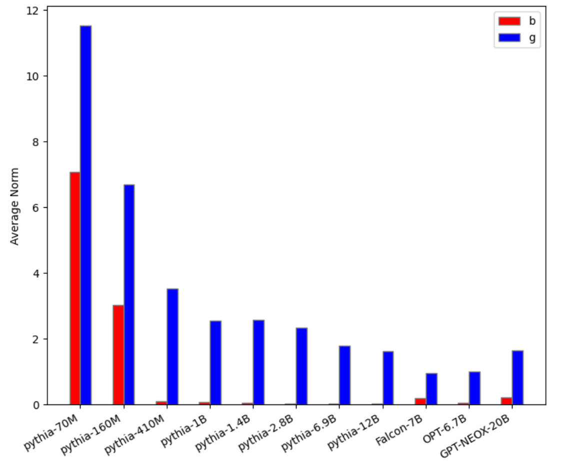


Figure 8: Comparisons of the average norm of the parameters $\mathbf{b}$ and $\mathbf{g}$ from Equation 1 for each of our models.

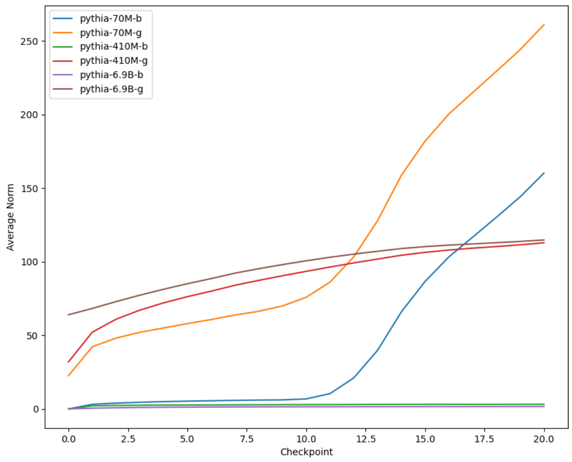


Figure 9: Comparisons of the average norm of the parameters $\mathbf{b}$ and $\mathbf{g}$ from Equation 1 across 21 evenly spaced checkpoints from training.

# 5 Discussion

## 5.1 Large Pythia Models Mitigate the Representation Degradation Problem

We have seen across numerous scales and with multiple metrics that Pythia models are the most isotropic across all of our and previous work. Pythia models contextualize words well, for instance, the 6.9B model has an *Inter-Sim* of 0.14 which corresponds to an angle of 81.6°, and an average *Intra-Sim* of 0.50 meaning tokens are well contextualized, as a *Intra-Sim* value close to 1 or 0.14 would represent poor contextualization. Considering the bias towards frequent tokens when calculating *Inter-Sim*, and the effect of Layer Norm it may be harder to get significantly lower values than these.

## 5.2 Degrading to Anisotropy Doesn't Happen Continually During Training

Gao et al. (2019) prove that the solution to the general optimization problem of the loss in Equation 2 is in the direction of a vector $v$ such that $\langle v, H_L(s,i)\rangle < 0$ for all $s$ and $i$, called a uniformly negative direction, and that as the last layer of the model is the Layer Norm, this $v$ exists under a very likely restriction

$$\sum_{i=1}^{d} \frac{b_i}{g_i} \neq 0 \tag{9}$$

where $g_i$ and $b_i$ are from $\mathbf{g}$ and $\mathbf{b}$ in Equation 1. However, this is the general optimization solution, not necessarily the solution that gradient batch optimization finds. Biś et al. (2021) show that the actual update per hidden state under gradient descent is

$$\mathbf{W}' = \mathbf{W} - \delta H_L(s,i)^{\intercal}\mathbf{y} + \delta H_L(s,i)^{\intercal}\hat{\mathbf{y}} \tag{10}$$

where $\delta$ is the learning rate, $\hat{\mathbf{y}}$ is the one-hots true label, and $\mathbf{y}$ are the predicated probabilities. In this sense, the words that are not the true label are pushed away from the hidden state. They call this the "common enemy problem".

First, we see that if the model is confident in its predictions, i.e., $||y - \hat{y}||_2$ is small, then the amount of change for each word is small. Secondly, as we are optimizing in batches, if we assume that our space of hidden states is isotropic then the "common enemies" work against each other, causing a potentially neutral change in isotropy. Lastly, Equation 10 is a simplification, as most Transformer models are trained with an Adam optimizer (Kingma and Ba, 2014) which has separate update weights for each parameter. All these things mean it is hard to determine the true effect of training on isotropy. We see in Figures 3, 4, and 7 that no model shows a steady decrease to anisotropy. The Pythia-70M model, which ends its training in an anisotropic state, shows an increase in isotropy for nearly the first half of training. It should be noted, that if we assume we have a highly anisotropic space then "common enemies" do work together, as we see when the Pythia-70M's anisotropy quickly "snowballs" during the last half of training. What causes this initial drop in isotropy is still unclear.

### 5.3 Large Pythia Models Optimize the Final Layer Norm for Isotropy

Looking at Equation 1, normalizing $\mathbf{h}$ with respect to mean and standard deviation maps $\mathbf{h}$ to the intersection of the unit ball and the hyperplane with normal $\overrightarrow{1}$. Multiplying by $\mathbf{g}$ maps points to the hyperplane with normal $\mathbf{g}' = (\frac{1}{g_1}, \ldots, \frac{1}{g_d})$. This means, even if $\mathbf{b} = \overrightarrow{0}$, that the space will look anisotropic using the $I()$ function. However, the points in the hyperplane may be otherwise isotropic as we see with our *Inter-Sim* and *Intra-Sim* analysis.

Gao et al. (2019) show when Equation 9 is true that all hidden states created by the layer norm lie on one side of the hyperplane with normal $\mathbf{g}'$. Another way to think of this is there is a rotation matrix such that the space of hidden states all have a positive value in the first dimension. As cosine similarity is rotation invariant, this shared positive dimension puts a positive lower bound on the *Inter-Sim* calculation if the space is otherwise isotropic. The impact of these shared positive values is proportional to the parallel portion of $\mathbf{b}$ with respect to $\mathbf{g}'$ and is minimized if this parallel portion has low norm. The perpendicular portion of $\mathbf{b}$ with respect to $\mathbf{g}'$ can also cause isotropy by shifting the space in a shared common direction, and this shift is minimized if the perpendicular portion has low relative norm compared to $\mathbf{g}$.

Looking at Figure 8, we see that all isotropic Pythia models minimize the norm of $\mathbf{b}$ generally and with respect to $\mathbf{g}$, and that the anisotropic Pythia models fail to do either. We also see that Pythia models, the most isotropic under all our metrics, are also the best across all models at this optimization. In fact, looking at Figures 1 and 2, we see that the final Layer Norm for said models, despite its potential for anisotropy, actually increased isotropy compared to the previous layer. Previous work has taken it as assumed that this would not happen during typical optimization (Gao et al., 2019).

### 5.4 Transitions to Anisotropy Correlate with Decreased Performance

Previous work has shown that the Pythia-70M model has worsening performance on generative tasks correlating with the decrease in isotropy (Biderman et al., 2023). We confirm this also applies to classification tasks, using SentEval with the default parameters (Conneau and Kiela, 2018), and find similar results, reported in Appendix A.

### 5.5 Not Tying Embedding Weights Increases Isotropy for Large Models

We see our most isotropic models, all large Pythia models and GPT-NeoX-20B, have separate embedding and unembedding weights. We also note, that the cost of untying weights for large models is quite small: 4.2% for Falcon-7B, 3.1% for OPT-6.7B, 1.5% for GPT-NEOX-20B, 2.5% for Llama-2-7B, and 0.4% for Llama-2-70B (Touvron et al., 2023). Our results are also in line with previous work, which showed that tying weights in small models, where the additional parameter cost is high (e.g., 50% increased parameters), improves performance (Press and Wolf, 2017; Inan et al., 2017), even though the Pythia-70M and Pythia-160M models have the worst isotropy across all models. Untying weights also has interpretability benefits (Belrose et al., 2023) and models have good performance dropping the unembedding matrix completely (Godey et al., 2023a).

## 6 Conclusions

We have found a strong negative result that the anisotropy of Transformer models can be assumed. We show that large Pythia models are isotropic across all large model sizes using numerous metrics. We find a correlation between having untied embedding and unembedding matrices and high isotropy, and show that, contrary to previous assumptions, Pythia models in fact optimize the final Layer Norm operation for isotropy. We have also explored how isotropy changes during training across different model scales. This work, providing a set of contrasting points, is a good first step into a deeper understanding of isotropy and its impacts.

Future work should consider an analysis of bias (Fuster Baggetto and Fresno, 2022) and clustering (Cai et al., 2021) for these isotropic models, and a proper ablation study to confirm that untied embedding matrices is the root cause of this isotropy.

## 7 Ethics Statement

To the best of our knowledge this work has no ethical concerns. We also note that we are making no claims about increases in fairness or decreases in bias in the languages modeling task (Navigli et al., 2023) or in frequency based bias seen when representation distort (Zhou et al., 2021).

## 8 Limitations

While we have added non-Pythia models to our analysis as comparative points and compare against previous work, these comparisons are not a substitution for a proper ablation study. In fact, the results for the GPT-NeoX-20B suggest such an ablation study is needed. While it has the next best results after the Pythia models, those result are not in line with the Pythia models. This is surprising as the architecture, datasets, and training of GPT-NeoX-20B are quite similar to Pythia models.

We have shown that models that end training in an anisotropic state do not always steadily tend towards this anisotropic state as previous works assumed. Instead we see a rise in isotropy followed by a drop and a runaway anisotropic effect. While we have provided reasoning for why the steady tend to anisotropy doesn't happen and why the runaway effect does, it is an open question as to why the phase change from isotropic to anisotropic begins in the first place and future work could explore this using the Pythia model training checkpoints.

We shown that large Pythia models optimize their final Layer Norm operation for isotropy, but have only shown this empirically. We provide no theoretical reasoning as to why this optimisation happens for large Pythia models and not for other large models. Further, we make no claims about the cause and effect relations between the final Layer Norm parameters and the isotropy of the unembedding matrix beyond our empirical observations.

We have only made claims regarding token embeddings. While it is unlikely that a space of isotropic token embeddings leads to an highly anisotropic space of sentence embeddings we did not have room to include a proper analysis to confirm this.

These isotropic Transformer models are autoregressive model, to our knowledge there is still no globally isotropic example for models trained using Masked Language Modeling such as BERT (Devlin et al., 2019).

Four days before the submission deadline a model with 360 checkpoints came out that has untied embeddings and unembedding matrices, does not use Layer Norm as it's final operation, and has poor isotropy compared to the Pythia models.[1]

[1] https://www.llm360.ai/blog/introducing-llm360-fully-transparent-open-source-llms.html

## References


Mira Ait-Saada and Mohamed Nadif. 2023. Is anisotropy truly harmful? a case study on text clustering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1194–1203, Toronto, Canada. Association for Computational Linguistics.

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2016. A Latent Variable Model Approach to PMI-based Word Embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *CoRR*, abs/2303.08112.

Stella Biderman, Kieran Bicheno, and Leo Gao. 2022. Datasheet for the Pile. *arXiv e-prints*, page arXiv:2201.07311.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Daniel Biś, Maksim Podkorytov, and Xiuwen Liu. 2021. Too much in common: Shifting of embeddings in transformer language models and its implications. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5117–5130, Online. Association for Computational Linguistics.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large scale autoregressive language modeling with Mesh-Tensorflow.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usvsn Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-NeoX-20B: An open-source autoregressive language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136, virtual+Dublin. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. 2021. Isotropy in the contextual embedding space: Clusters and manifolds. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Re. 2022. Flashattention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*.

David Demeter, Gregory Kimmel, and Doug Downey. 2020. Stolen probability: A structural weakness of neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2191–2197, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yue Ding, Karolis Martinkus, Damian Pascual, Simon Clematide, and Roger Wattenhofer. 2022. On isotropy calibration of transformer models. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pages 1–9, Dublin, Ireland. 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.

Alejandro Fuster Baggetto and Victor Fresno. 2022. Is anisotropy really the cause of BERT embeddings not being semantic? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4271–4281, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The Pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nathan Godey, Éric de la Clergerie, and Benoît Sagot. 2023a. Headless Language Models: Learning without Predicting with Contrastive Weight Tying. *arXiv e-prints*, page arXiv:2309.08351.

Nathan Godey, Éric de la Clergerie, and Benoît Sagot. 2023b. Is Anisotropy Inherent to Transformers? In *Poster at 61st Annual Meeting Student Research Workshop*, Toronto, Canada. Association for Computational Linguistics.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2017. Tying word vectors and word classifiers: A loss framework for language modeling. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Olga Kovaleva, Saurabh Kulshreshtha, Anna Rogers, and Anna Rumshisky. 2021. BERT busters: Outlier dimensions that disrupt transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3392–3405, Online. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *arXiv e-prints*, page arXiv:1607.06450.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130, Online. Association for Computational Linguistics.

Ziyang Luo, Artur Kulmizev, and Xiaoxi Mao. 2021. Positional artefacts propagate through masked language model embeddings. 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 5312–5327, Online. Association for Computational Linguistics.

Jiaqi Mu and Pramod Viswanath. 2018. All-but-the-top: Simple and effective postprocessing for word representations. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Roberto Navigli, Simone Conia, and Björn Ross. 2023. Biases in large language models: Origins, inventory, and discussion. *J. Data and Information Quality*, 15(2).

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 157–163, Valencia, Spain. Association for Computational Linguistics.

Giovanni Puccetti, Anna Rogers, Aleksandr Drozd, and Felice Dell'Orletta. 2022. Outlier dimensions that disrupt transformers are driven by frequency. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1286–1304, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sara Rajaee and Mohammad Taher Pilehvar. 2021. A cluster-based approach for improving isotropy in contextual embedding space. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 575–584, Online. Association for Computational Linguistics.

Sara Rajaee and Mohammad Taher Pilehvar. 2022. An isotropy analysis in the multilingual BERT embedding space. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1309–1316, Dublin, Ireland. Association for Computational Linguistics.

Noam Shazeer. 2019. Fast Transformer Decoding: One Write-Head is All You Need. *arXiv e-prints*, page arXiv:1911.02150.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

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.

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. *arXiv e-prints*, page arXiv:2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Dilin Wang, Chengyue Gong, and Qiang Liu. 2019. Improving neural language modeling via adversarial training. In *International Conference on Machine Learning*, pages 6555–6565. PMLR.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2020a. Improving neural language generation with spectrum control. In *International Conference on Learning Representations*.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2020b. Improving neural language generation with spectrum control. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. ConSERT: A contrastive framework for self-supervised sentence representation transfer. 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 5065–5075, Online. Association for Computational Linguistics.

Sangwon Yu, Jongyoon Song, Heeseung Kim, Seongmin Lee, Woo-Jong Ryu, and Sungroh Yoon. 2022. Rare tokens degenerate all tokens: Improving neural text generation via adaptive gradient gating for rare token embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29–45, Dublin, Ireland. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zhong Zhang, Chongming Gao, Cong Xu, Rui Miao, Qinli Yang, and Junming Shao. 2020. Revisiting representation degeneration problem in language modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 518–527, Online. Association for Computational Linguistics.

Kaitlyn Zhou, Kawin Ethayarajh, and Dan Jurafsky. 2021. Frequency-based Distortions in Contextualized Word Embeddings. *arXiv e-prints*, page arXiv:2104.08465.

# A Appendix: SentEval Classification Tasks During Training

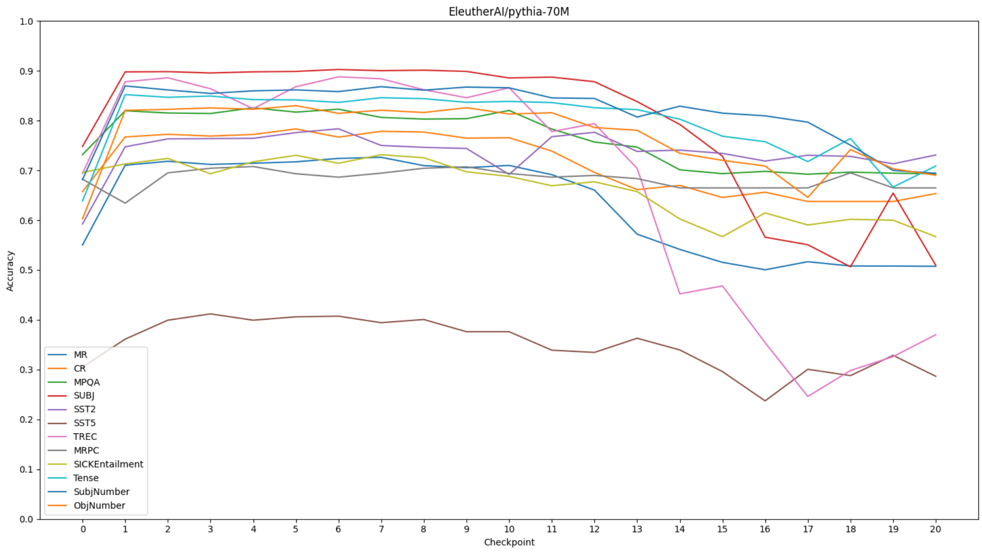


Figure 10: Accuracy on classification tasks for the Pythia 70M model.

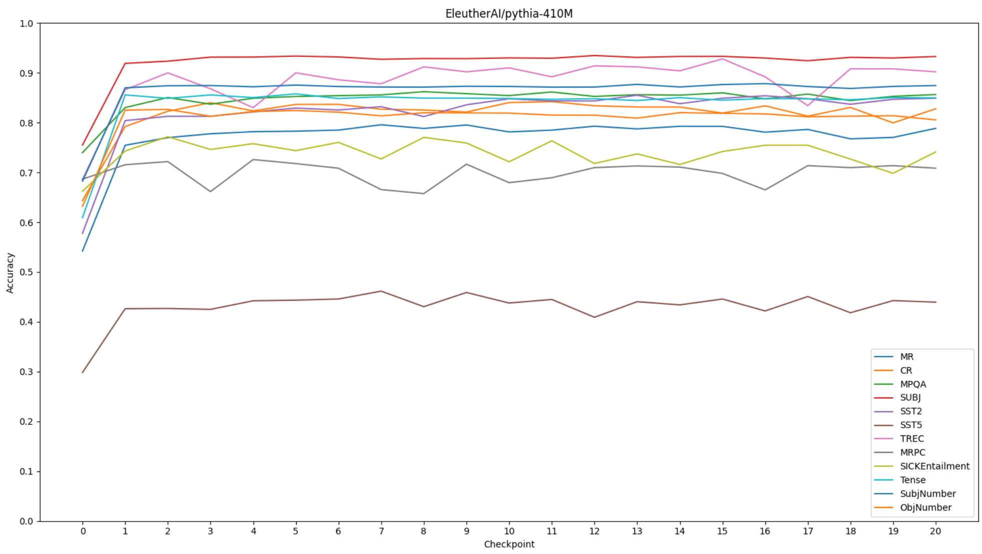


Figure 11: Accuracy on classification tasks for the Pythia 410M model.

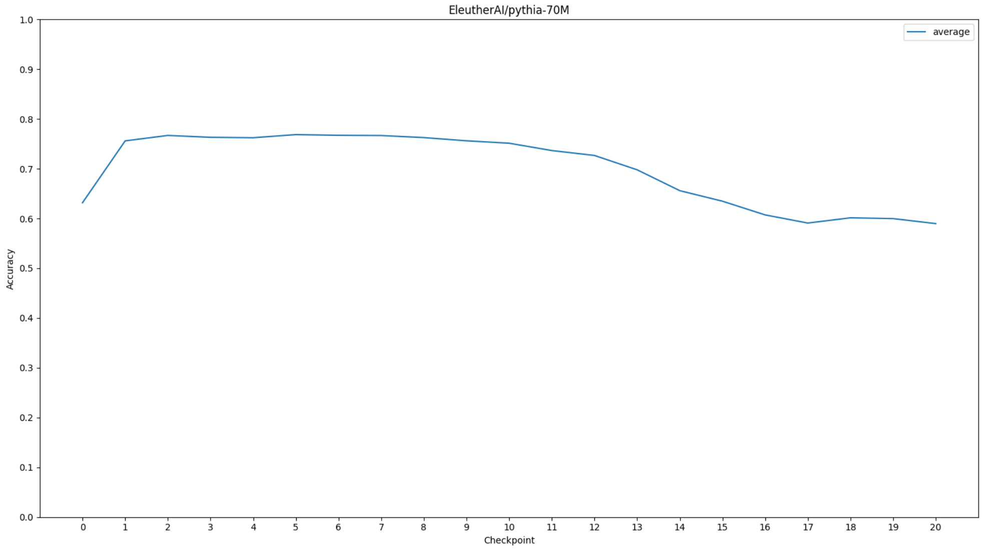


Figure 12: Average accuracy on classification tasks for the Pythia 70M model.

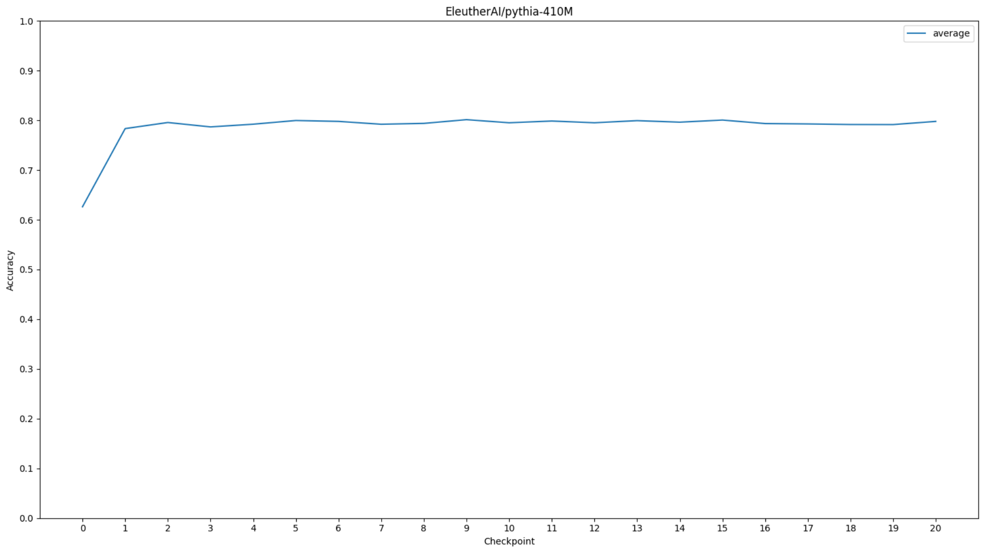


Figure 13: Average accuracy on classification tasks for the Pythia 410M model.

## B Appendix: Datasets and Training

| | 70M | 160M |
|---|---|---|
| Enron Emails | 0.27 | 1.08 |
| NIH Exporter | 0.92 | 3.62 |
| PhilPapers | 1.45 | 5.79 |
| HackerNews | 2.54 | 10.21 |
| EuroParl | 3.70 | 14.48 |
| Ubuntu IRC | 4.38 | 17.27 |
| DM Mathematics | 8.95 | 37.00 |
| Wikipedia (en) | 9.72 | 38.58 |

Table 1: Computation times in hours using a 1080TI

| | 70M | 160M | 410M | 1.4B |
|---|---|---|---|---|
| Enron Emails | 0.06 | 0.22 | 0.53 | 1.23 |
| NIH Exporter | 0.22 | 0.71 | 1.76 | 4.53 |
| PhilPapers | 0.35 | 1.17 | 2.81 | 7.25 |
| HackerNews | 0.61 | 2.09 | 4.94 | 12.78 |
| EuroParl | 0.87 | 2.68 | 7.00 | 18.21 |
| Ubuntu IRC | 1.03 | 3.26 | 8.48 | 21.25 |
| DM Mathematics | 2.08 | 7.89 | 17.31 | - |
| Wikipedia (en) | 2.28 | 7.67 | 19.00 | - |

Table 2: Known computation times in hours using an A100

| Source | Processed Size (GiB) | Mean Document Size (KiB) | Sentences | Tokens |
|---|---|---|---|---|
| Enron Emails | 0.46 | 1.78 | 3206547 | 107063699 |
| NIH Exporter | 2.00 | 2.11 | 11402784 | 376537632 |
| PhilPapers | 2.40 | 73.37 | 18172474 | 584403514 |
| HackerNews | 4.20 | 4.92 | 36334985 | 1024155017 |
| EuroParl | 6.40 | 68.87 | 30033886 | 1519805406 |
| Ubuntu IRC | 6.70 | 545.48 | 33988454 | 1741293414 |
| DM Mathematics | 8.40 | 8.00 | 171791406 | 3573649454 |
| Wikipedia (en) | 18.10 | 1.11 | 121580702 | 3920248990 |

Table 3: Dataset information for sources used in our analysis.