BERT Busters: Outlier Dimensions that Disrupt Transformers

Olga Kovaleva[*1], Saurabh Kulshreshtha[*1], Anna Rogers[2] and Anna Rumshisky[1]
[1]Department of Computer Science, University of Massachusetts Lowell
[2]Center for Social Data Science, University of Copenhagen
[1]{okovalev,skul,arum}@{cs.uml.edu}
[2]arogers@sodas.ku.dk

Abstract

Multiple studies have shown that Transformers are remarkably robust to pruning. Contrary to this received wisdom, we demonstrate that pre-trained Transformer encoders are surprisingly fragile to the removal of a very small number of features in the layer outputs (<0.0001% of model weights). In case of BERT and other pre-trained encoder Transformers, the affected component is the scaling factors and biases in the LayerNorm. The outliers are high-magnitude normalization parameters that emerge early in pre-training and show up consistently in the same dimensional position throughout the model. We show that disabling them significantly degrades both the MLM loss and the downstream task performance. This effect is observed across several BERT-family models and other popular pre-trained Transformer architectures, including BART, XLNet and ELECTRA; we also show a similar effect in GPT-2.

1 Introduction

Pre-trained Transformer-based models (Vaswani et al., 2017) have become widely popular in a variety of NLP applications. Multiple studies of BERT-family models (Devlin et al., 2019) showed that Transformers are remarkably robust to pruning (Gordon et al., 2020; Prasanna et al., 2020; Chen et al., 2020; Michel et al., 2019). This work presents a different and unexpected result: it is possible to dramatically disrupt the performance of BERT and other Transformer-based architectures by modifying very few weights (less than 0.0001% for BERT).

In particular, we show that there is a very small number of outlier dimensions that regularly appear in the same position in the pre-trained encoder layers of a given Transformer model. We demonstrate that this effect holds for different Transformer-family architectures, including multiple variants of BERT, as well as ELECTRA (Clark et al., 2020), BART (Lewis et al., 2020), and XLNet (Yang et al., 2019). A similar phenomenon is also present in the decoder layers of GPT-2 (Radford et al.). When these dimensions are disabled throughout the model in the concluding transformation of each layer, they can drastically reduce the overall model performance. With the exception of GPT-2, the last transformation in each layer of these models is normalization (LayerNorm), which is what we mainly focus on in this study.

The contributions of this work are as follows:

- We identify certain outlier dimensions in Transformer layer outputs and show that they play a crucial role in both language modeling and downstream task performance. Disabling the weights for these output dimensions drastically degrades performance (up to 44 points).
- We show that this effect holds for the encoder layers of six different models of the BERT family, as well as other popular pre-trained Transformer-based models including ELECTRA, BART, and XLNet. In GPT-2, a similar phenomenon is observed in the output dense transformation of the decoder layers.
- We demonstrate that outlier weights emerge gradually and begin to emerge early in pre-training, causing abnormal spikes at select dimensions in the output embedding vectors.

To our knowledge, this is the first work to establish the presence of very few regular outliers in the output Transformer representations and their importance for the model performance. It is not clear why these features emerge, but the final transformations clearly play a larger role in the Transformer layers than is usually assumed, and this needs further investigation.

[*] Authors contributed equally to this work.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 3392–3405

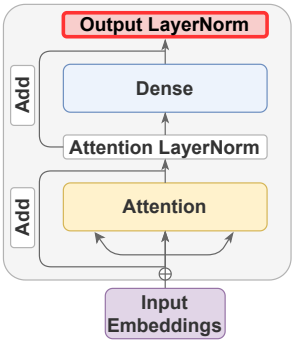

Figure 1: Transformer encoder layer, adapted from (Vaswani et al., 2017).

This paper is organized as follows. After a brief overview of related work (§2), we introduce the methodology for defining, locating, and disabling the BERT outlier weights in §3. In §4.1 and §4.2, we quantify the effect of disabling these weights both in pre-training and in downstream tasks. In §4.3, we demonstrate that other Transformers (BART, ELECTRA, XLNet, and GPT-2) also exhibit similar behavior. §5.1 evaluates magnitude- and position-based criteria for identifying the outlier dimensions and compares them with our proposed criteria. In §5.2, we replicate the outlier effect in a BERT model during pretraining and study its dynamics.

2 Related work

Transformer layer outputs. At a high level, Transformer encoder layers consist of multi-head self-attention followed by a dense layer Vaswani et al. (2017). Most contemporary Transformers use normalization to improve the speed and stability of training.

Usually, the outputs of both self-attention and linear layers undergo the layer normalization transformation (LayerNorm, Ba et al. (2016)). Each LayerNorm transformation is parameterized by a separate set of learned weights (*scaling factors* and *biases*). Xiong et al. (2020) refer to this configuration as *post-LN*. In the *pre-LN* variant adopted by the GPT-2 model, LayerNorm is applied prior to the self-attention or linear transformations instead.

We will refer to the outputs of the final transformation in the encoder layer as *features* and the parameters of this transformation as *weights*. The final transformation is LayerNorm for all models considered in this study except GPT-2, where the last component is a MLP.

Like other normalization techniques, LayerNorm operates in two steps. For a given input x_i of the i-th layer with a hidden dimension m, LayerNorm computes mean and variance across the features:

$$\mu_i = \frac{1}{m}\sum_{j=1}^{m} x_{ij},\ \sigma_i^2 = \frac{1}{m}\sum_{j=1}^{m}(x_{ij}-\mu_i)^2 \quad (1)$$

The inputs are then normalized and a learnable scale-shift transformation is applied to produce the normalized output embedding:

$$\hat{x}_{ij} = \frac{x_{ij}-\mu_i}{\sqrt{\sigma_i^2+\epsilon}},\ y_i = \gamma \odot \hat{x}_i + \beta \quad (2)$$

where $\gamma \in \mathbb{R}^m$ and $\beta \in \mathbb{R}^m$ are trainable parameters referred to as *scaling factor* and *bias* (shift).

So far there have been few studies of normalization strategies in Transformer architectures and they focused mostly on the description of the training process. Xiong et al. (2020) show that the location of LayerNorm in Transformer affects the gradient flow and demonstrate the need of the warmup stage. Nguyen and Salazar (2019) inject multiple normalization blocks in specific network submodules to improve model performance. More recently, LayerNorm alternatives have been proposed and shown to have better gradient propagation through the network (Xu et al., 2019; Shen et al., 2020).

Overparametrization. After the initial reports of redundancy in the BERT model (Michel et al., 2019; Kovaleva et al., 2019), compressing Transformers quickly became a subfield of its own (Jiao et al., 2020; Zafrir et al., 2019; Fan et al., 2019; Guo et al., 2020). See overviews by Ganesh et al. (2020) and Rogers et al. (2020).

Pruning is a class of methods for model compression which involves setting some of its weights to zeros with minimal performance loss. Much pruning work focuses on compression for the sake of efficiency, but it is also used for model analysis, and that is our goal as well. The most common approach is selecting the weights to be pruned by magnitude (Han et al., 2015).

Some of the recent findings are that the lottery ticket hypothesis (Frankle and Carbin, 2019) holds for BERT: its largest weights do form subnetworks that can be retrained alone to reach the performance close to that of the full model (Prasanna et al., 2020; Chen et al., 2020; Gordon et al., 2020). In structured pruning, the best subnets of BERT's heads and MLPs (selected by importance scores)

do not quite reach the full model performance, but the worst ones are still much better than the worst magnitude-based subnets (Prasanna et al., 2020), presumably because they retain a lot of high-magnitude weights.

3 Outlier weights in BERT models

In this section, we introduce the methodology for identifying outlier dimensions and describe our method for disabling these outlier features.

3.1 Identification

To identify the outlier weights in BERT-like models, we consider all the output components in each encoder layer. We compute the mean and standard deviation of the bias and scaling factors of the output LayerNorm. We identify the dimensions where both of these weights are at least 3σ from the mean. Figure 2 illustrates this heuristic. Further, we select the dimensions where this is consistently the case for at least half of the model layers. We refer to these dimensions as outliers.

The described heuristic was used to identify the outlier dimensions in four out of six BERT models we considered: BERT-base, BERT-medium, BERT-small and mBERT.[1] For BERT-large, the deepest model we considered, the frequency constraint was relaxed to 1/3 of the layers. In RoBERTa (Liu et al., 2019), the distribution of the scaling factors was a little different from BERTs, and we relaxed the standard deviation constraint down to 2 sigmas to detect the outliers. In Section 10.1 of Appendix, we report positions of outlier weights identified for all models.

3.2 Disabling

To quantify the effect of the outlier weights on BERT, we disable them and examine how this affects model performance. We set the outlier weights to zeros across all layers and report model performance on a) masked language modeling and b) downstream GLUE tasks (Wang et al., 2018).

Since different model components may affect performance, we also looked at all the parameter

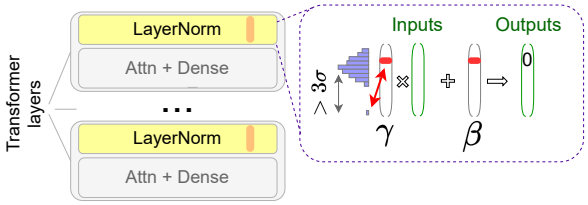

Figure 2: Simplified illustration of our approach to disabling LayerNorm weights. We consistently set the outlier weights γ and β of the output LayerNorm to zeroes, which results in masking of the corresponding features in the output vectors. We repeat the procedure for all of the Transformer layers of the encoder.

vectors and matrices in BERT that have the same dimensionality as the output embeddings. These included key, query and value transformations[2], output LayerNorm, attention LayerNorm, and input embedding layers. In order to examine the effect of these components on model performance, we masked the identified outlier weights of a given component simultaneously across all layers at the same dimensional position. When working with the matrices, we set to zeros the entire row of weights corresponding to the outlier positions across all Transformer layers. Similarly, for LayerNorm, we set the scaling factor and the bias at the outlier position across all Transformer layers to zero. In both cases, this results in "masking" of the output vector's feature at the specified dimension after a forward pass through that layer.

Although the same dimensions repeatedly show up as outliers across different model components, in the preliminary experiments, we found that disabling the weights of the input embedding layers and of the linear layers produced no significant change in performance or in the output embedding space, so we did not pursue this direction further. However, the outlier weights of the output LayerNorm had an unexpectedly large effect on the model, and this is what we focus on in most of our experiments.

3.3 Visualization

As an example, let us consider the two outlier dimensions that the above method identifies for BERT-base-uncased model: `308` [3] and `381`. Fig-

[1] BERT-medium and BERT-small come from the official Google repository (https://github.com/google-research/bert), and the other models from the HuggingFace (https://github.com/huggingface/transformers). Interestingly, we discover that the checkpoints of the same BERT-base configuration provided by different repositories (Huggingface vs. Google) have the outliers in different locations; the outliers also have different values: positive vs. negative.

[2] To find outliers in weight matrices, we compute the L_1 norm for each row. The total number of rows is the same as the dimensionality of the layer output embedding (e.g. 768 for BERT-base). From this distribution of row-norms, we select those row indices for which the magnitude is 3σ away from the mean of the distribution.

[3] All dimensions in the paper are zero-indexed.

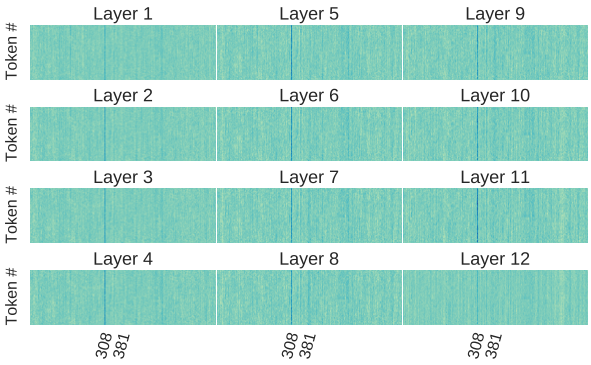

Figure 3: Outlier LayerNorm features `308`, `381` in BERT-base-uncased (randomly sampled input).

ure 3 shows a heatmap of the output embedding of each layer (one pixel per row) for a random passage from WikiText (Merity et al., 2016). Since the output embeddings are produced by LayerNorm, the outlier dimensions with unusually high or low-magnitude weights should be visible in the heatmap.

As seen in Figure 3, dimension `308` consistently produces high-magnitude weights in the output embeddings in most BERT layers; feature `381` shows visibly high values in layers 7-10. The magnitude of a given feature depends both on the LayerNorm scaling factor and the bias (Equation 2). We find that both contribute to the outlier effect, to various degrees in different layers. See Table 13 in the Appendix for statistics on all scaling factors and biases in BERT-base.

Features that show high magnitude throughout the model are expected to distort the resulting embedding space. In a brief experiment, we found that the vector representations of up to 95% of the input tokens from the WikiText corpus have abnormally high magnitudes at the dimensions corresponding to the outlier weights we identified. We found that this embedding distortion is not attributable to the input embedding layer (Layer 0). We did this by manually setting select embedding weights of the three channels of the input embedding layer (token, token type, and position) to zero (along with the weights of the following normalization layer).

The fact that the embedding distribution we observed is not uniform is in line with observations by Ethayarajh (2019) who concluded that BERT embeddings are highly anisotropic and form a cone-like shape in the hidden space. The outlier weights we identify are likely the cause of this, since after they are removed, the embedding space becomes relatively uniform.

Input	Ghostbusters was [released] on June 8 , [1984] , to critical [acclaim] and became a cultural phenomenon . It was well [received] for its deft blend of comedy, [action] , and horror , and Murray ' s performance was [repeatedly] singled out for praise .
RoBERTa	Ghostbusters was [released] on June 8 , [1986] , to critical [acclaim] and became a cultural phenomenon . It was well [received] for its deft blend of comedy, [action] , and horror , and Murray ' s performance was [often] singled out for praise .
Random	Ghostbusters was [released] on June 8 , [1986] , to critical [acclaim] and became a cultural phenomenon . It was well [received] for its deft blend of comedy, [action] , and horror , and Murray ' s performance was [particularly] singled out for praise .
Outliers	{ lock was [never] on June 8 , [</s>] , to rely [,] and . It was well [known] for its acker of comedy , [dinner], and horror , and Murray ' s was [ever] , </s> </s>)

Table 1: Input masked tokens (blue) are given in brackets. RoBERTa correctly reconstructs 4 out of 6 masked tokens (green), and fills in plausible (brown) predictions for the remaining 2 tokens. RoBERTa with 2 randomly disabled LayerNorm dimensions works almost the same as the base model. However, RoBERTa with 2 outlier LayerNorm dimensions makes incorrect and implausible (red) predictions, and changes the hidden token states significantly enough to map the unmasked input tokens to other, often non-sensical words. In this example, we do not show the special tokenizer tokens.

4 Effects of Disabling Outlier Weights

In this section, we consider the effects of disabling outlier weights in BERT on language modeling (§4.1) and on downstream tasks (§4.2). We also investigate whether other Transformers exhibit a similar phenomenon (§4.3).

4.1 Masked Language Modeling

Our key finding is that disabling the outlier dimensions significantly degrades the quality of the language model, even though fewer than 0.001% weights of the model are affected. Table 1 shows a sample output before and after the LayerNorm outliers are disabled in RoBERTa[4]. It is clear that the quality of the language model degrades dramatically, while disabling an equal number of randomly selected non-outliers has almost no effect.

To quantify this effect, we measure the cross-entropy loss before and after disabling each dimension of the LayerNorm on at a time. We do this on the validation subset of the WikiText corpus

[4]More sample outputs are given in the Appendix.

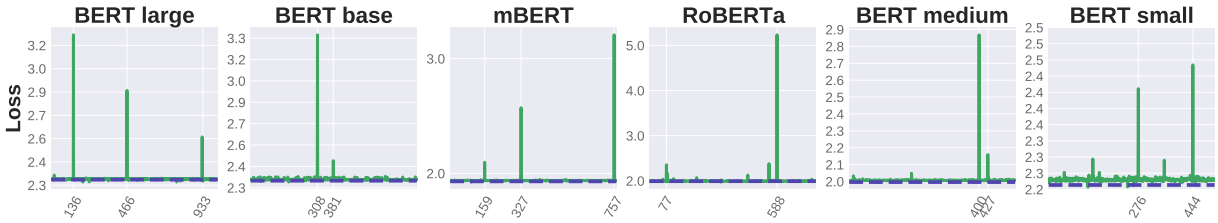

Figure 4: Language model cross-entropy loss (green) after the scaling factor and bias for a given dimension are set to zero, one at a time. The dashed blue line shows the loss achieved by the full model.

		BERT-large		BERT-base		mBERT		RoBERTa		BERT-medium		BERT-small	
		#w	CE	#w	CE	#w	CE	#w	CE	#w	CE	#w	CE
	Baseline	0	2.28	0	2.30	0	1.93	0	1.99	0	2.00	0	2.26
Single	Random	48	2.29	24	2.31	24	1.95	24	2.03	16	2.01	8	2.26
Single	Top outlier	48	3.22	24	3.33	24	3.21	24	5.23	16	2.87	8	2.44
All	Random	144	2.29	48	2.31	72	1.96	48	2.00	32	2.04	16	2.28
All	All outliers	144	**5.49**	48	**4.53**	72	**6.92**	48	**7.85**	32	**3.21**	16	**2.93**

Table 2: Cross-entropy (*CE*) on the validation set of WikiText when the LayerNorm weights are zeroed. *Single* shows the performance when the most damaging outlier is disabled vs. disabling one non-outlier feature (averaged over all non-outliers disabled one at a time). *All* shows performance for when all outliers in a given model are disabled vs. disabling an equal number of randomly selected non-outliers (averaged over 1000 runs). *#w* indicates the total number of modified weights in a given model.

(Merity et al., 2016). We use the standard maximum sequence length of 256 and the token masking probability of 0.15.

All tested models show surprising sensitivity to zeroing out the weights at the outlier positions across all layers of the model (Figure 4). For example, removing only 24 parameters (the scaling factor and the bias of a specific LayerNorm dimension across all 12 layers) increases RoBERTa's loss by almost a factor of 4.

Table 2 shows even more drastic effects when scaling factors and biases for all the outlier dimensions are disabled simultaneously. For comparison, we randomly sample an equal number of non-outlier dimensions and disable the corresponding LayerNorm weights throughout the model. We report the loss averaged over 1000 runs.

4.2 BERT Downstream Tasks

In order to investigate the effect of outlier weights on downstream performance, we evaluate BERT-base on the GLUE benchmark tasks (Wang et al., 2018), with the exclusion of Winograd Schema Challenge, which BERT generally fails to learn (Prasanna et al., 2020). We use the evaluation split of the GLUE benchmark for which the labels are publicly available. As described above, BERT has two outlier dimensions, `308` and `381`. We consider the following two sets of experiments:

1. Disable post fine-tuning. We fine-tune BERT on every GLUE task, then disable the outlier LayerNorm parameters (scaling factor and bias pair) across all layers as described in subsection 3.2. We experiment with disabling each of the two detected outliers both individually and simultaneously in pre-trained BERT-base. We compare the resulting performance to (a) removing the LayerNorm weights for all other hidden dimensions one-by-one, and (b) randomly sampling pairs of non-outlier dimensions so as to disable them simultaneously.

2. Disable pre-fine-tuning. We disable the layer norm parameters for the outlier dimensions *prior* to fine-tuning. Our goal is to check whether the fine-tuning allows the transformers to recover the information from rest of the parameters.

For all the fine-tuning runs, we set the learning rate to 5e-5, batch size to 64 and train for 4 epochs across all the experiments. Since BERT performance varies a lot due to task-specific initialization (Dodge et al., 2020), we use the same initialization across all experiments.

Figure 5 shows model performance when LayerNorm outliers are disabled one at a time. Table 3 compares task-specific performance of the outlier-disabled and the full model for each task.

The main takeaway from the *post fine-tuning* experiments is that disabling one or the other of the

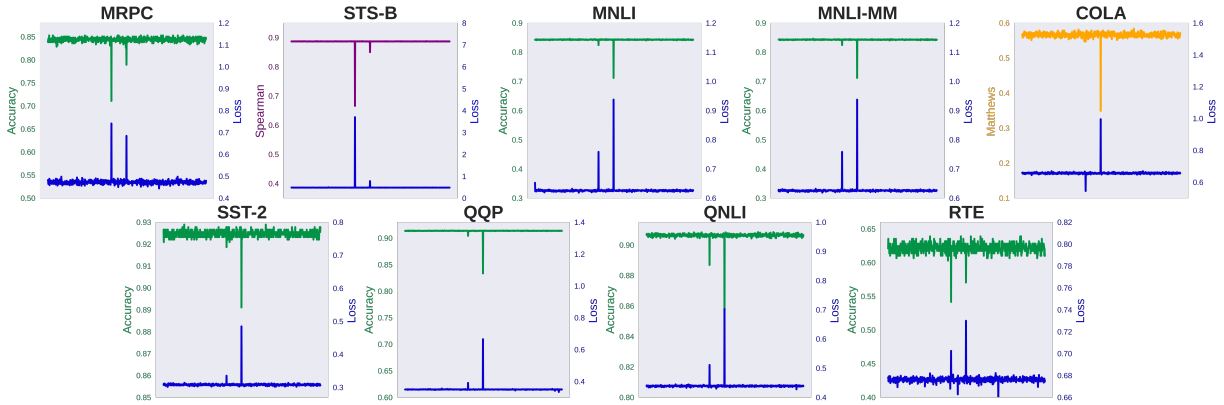

Figure 5: Performance of BERT-base on GLUE tasks when one LayerNorm weight at a time is disabled throughout the model. Dimensions are shown on the X axis. Loss (blue) and accuracy, or correlation coefficients, where applicable (other colors) are shown.

		MRPC	STS-B	MNLI	MNLI-mm	COLA	SST-2	QQP	QNLI	RTE
	Baseline (full model)	87.2	88.8	84.1	84.2	56.8	92.5	89.8	90.6	61.7
Post-ft	Non-outlier†	+0.3	-0.1	-0.2	-0.1	+0.2	0	-0.1	0	-0.4
	Outlier-308	-10.5	-23.4	-2.2	-1.8	-2.16	-0.6	-1.0	-1.9	-7.2
	Outlier-381	-4.6	-4.4	-13.7	-13.0	-22.2	-3.4	-10.8	-7.3	-5.0
	Random non-outlier pair‡	-1.1	0.0	+0.3	+0.2	-0.5	+0.1	+0.1	0	+0.5
	Outliers 308 + 381	**-8.6**	**-44.1**	**-27.9**	**-27.2**	**-32.3**	**-20.8**	**-13.0**	**-12.2**	**-10.0**
Pre-ft	Random non-outliers*	-0.3	-0.05	-0.2	-0.2	+0.9	-0.06	-0.2	-0.3	+0.6
	Outlier-308	+0.3	-0.9	-0.5	+1.7	-0.3	+0.7	-0.1	0	-5.1
	Outlier-381	-2.4	-0.7	-0.6	-0.5	-0.9	-1.2	-0.7	-1.4	+4.0
	Outliers 308 + 381	-1.1	-1.6	-1.4	-0.7	-2.9	-1.7	-0.7	-2.3	-0.7

Table 3: Performance of the pretrained BERT-base model vs. different configurations with disabled outlier weights: post fine-tuning (*post-ft*) and pre-fine-tuning (*pre-ft*). †We disable each of non-outlier dimension parameters one at a time and average over them. ‡For the random sampling of the pairs of non-outlier dimensions, we report averages over 1000 runs. *For the pre-finetuning experiment where random non-outlier parameters are disabled, we sample 10 non-outlier dimensions randomly and disable LayerNorm weights and biases for them across the entire model.

outlier dimensions (or both) drastically degrades model performance on downstream tasks. Which of them affect downstream performance the most is highly task-dependent. For example, the outlier dimension 308 has little effect on CoLA(-2.16) but a large effect on STS-B(-23.4), and for 381 it's the opposite. On SST-2, QNLI, RTE neither outlier drops the performance by over 10 points individually, but disabling them both has a strong adverse effect.

In general, disabling two outliers together causes more severe damage across the board than disabling a single outlier. The overall performance drop is task-specific. The most adversely affected tasks are STS-B (-44.1) and CoLA (-32.3), which are the regression tasks that also suffered the most in pruning experiments by Prasanna et al. (2020). However, MNLI and SST-2 are classification tasks, and both of them also lose over 20 points. Note that disabling random non-outlier dimensions (either alone, or in pairs) has negligible effect on performance across tasks. Disabling one outlier and one random non-outlier has the same effect as disabling a single outlier.

In *pre-fine-tuning* experiments, the question we ask is whether the model can recover from the handicap we introduce, and still learn the task. We expect it to mostly recover, since even randomly initialized BERT without any pre-training can be fine-tuned to solve GLUE tasks fairly well (Kovaleva et al., 2019). In this case, since most of the pretrained sub-networks are still accessible to the classifier, we expect to see a strong recovery close to the baseline. We find this to be the case, however, the model is more adversely affected: -1.1 on average when two outliers are disabled, compared to disabling 10 non-outliers (-0.31 on average).

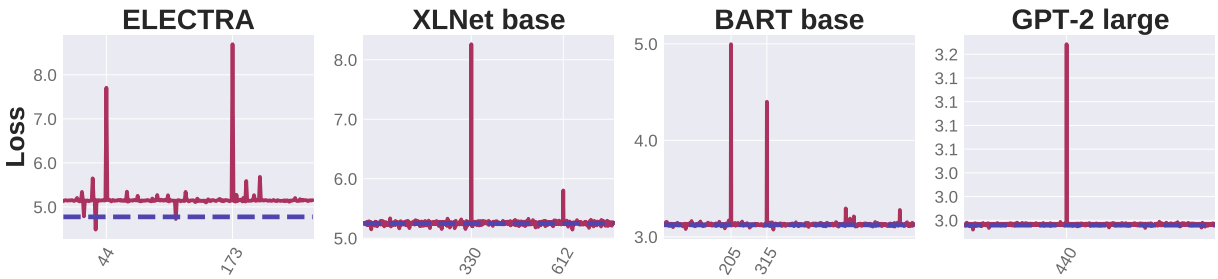

Figure 6: Performance (loss) of other Transformer models after a given LayerNorm scaling factor and bias are set to zero, one at a time. For GPT-2, the dense layer weights and biases are disabled, as it is an instance of a pre-LN model. The dashed blue line shows the loss for the full model.

Model	Task	Data	#Params	Disabled		Performance	
				# dims	weight %	Baseline	Disabled
ELECTRA	MLM	WikiText	110M	2	4×10^{-5}	4.8	**8.1**
XLNet	PLM	WikiText	120M	2	4×10^{-5}	5.2	**8.4**
BART	Summ.	CNN/Daily Mail	140M	2	9×10^{-6}	3.1	**4.4**
GPT2	CLM	WikiText	770M	1	0.024	3.0	**3.2**

Table 4: Loss increase in other Transformer models after layer normalization scaling factors and biases highlighted in Figure 6 are set to zero, compared to the baseline configuration. For GPT-2, the dense layer weights and biases are disabled instead, since it is a pre-LN model. *MLM*, *PLM*, and *CLM* stand for masked, permutation, and causal language modeling objectives, respectively. *Summ.* stands for the summarization task. *#dims* denotes the total number of dimensions modified.

4.3 Outliers in Other Transformers

Above, we described our methodology for identifying outliers in BERT models and studied how they affect model performance. In this section, we show that a similar phenomenon is observed in other popular Transformers: BART-base, ELECTRA-base generator, XLNet-base, and GPT-2 large.

Since our goal here was merely to confirm the existence of outlier dimensions, in these experiments, we simply disabled individual LayerNorm dimensions of the encoder part of the model across all Transformer layers. For GPT-2, we modify the weights of the dense output layer instead of the LayerNorm weights, since it uses the pre-LayerNorm (pre-LN) configuration (i.e., LayerNorm is placed before the output feature-producing feed-forward layer).

We perform this experiment for all models, measuring how this affects the loss function in the native pre-training tasks of three models: permutation language modeling task for XLNet, causal language modeling for GPT-2, and masked language modeling for the generator component of ELECTRA. For BART, we found that the pre-trained model had unusually high perplexity out of the box, and we substituted the modeling task with summarization on the CNN/Daily Mail dataset (Hermann et al., 2015).

As with BERT, our results (Figure 6) suggest that for each model, there are a few distinct dimensions which disrupt performance significantly more than the rest. We identify a few most impactful dimensions and also disable them at once, as reported in Table 4. The effect on perplexity is the least pronounced for GPT2, which we attribute to the fact that the model is significantly larger than the others, which may make it more robust to the disabling of individual weights. However, we found that when six dimensions are disabled simultaneously, the perplexity increases by over 300 times.

5 What Makes Outlier Weights Special

5.1 Magnitude or Location?

In this section, we conduct two experiments to validate our proposed criteria for selecting BERT weights to be disabled. Specifically, we want to understand if the same effects would be observed (1) with magnitude-based selection of LayerNorm parameters to be disabled, or (2) using our selection method, but disabling the outlier dimensions only in the first (input) layer or in the later layers (which may have a more direct effect on the output). We use the drop in the loss value for this comparison.

First, we compare the effect of disabling the

	CE	#w	CE	#w
baseline (full model)	2.30	0	2.30	0
Random	2.31	24	2.32	48
LSF	2.72	24	2.74	48
LB	3.21	24	3.42	48
Outlier-308	3.32	24	**4.53**	48
Outlier-381	2.44	24		

Table 5: BERT-base cross-entropy loss (*CE*) on the WikiText validation data when one (left) or two (right) outlier dimensions are disabled at a time, compared to magnitude-based pruning approaches (LSF and LSB). *#w* denotes the total number of modified weights.

selected dimensions (308 and 381 in BERT-base, disabled individually or together – i.e. disabling either 12 or 24 LayerNorm scaling factor and bias pairs) to disabling of the following alternatives:

- **Random**: disable 12 or 24 randomly selected pairs of LayerNorm scaling factor and bias pairs in the entire model;
- **Largest Scaling Factor (LSF)**: sort the LayerNorm scaling factors in the model by magnitude and disable the top 12 (or 24) scaling factors and the corresponding biases;
- **Largest Bias (LB)**: repeat the above using the LayerNorm biases instead, i.e. select the top 12 (or 24) LayerNorm biases and disable the corresponding scaling factor / bias pairs.

Table 5 suggests that simple magnitude-based pruning of the output LayerNorm results in a much smaller degradation. As compared to disabling both BERT outliers, the magnitude-driven pruning of the same number of weights results in the value of cross-entropy that is 1.3x smaller.

Looking at the effects of disabling the outlier dimensions only in a subset of layers, Table 6 shows that modifications made to the input layer have little to no effect. Interestingly, switching off the weights in the last Transformer layer, which is used for computing inputs for task-specific classifiers, also does not disrupt the model. However, as we begin to disable earlier layers and the number of layers with disabled weights increases, we observe progressively larger loss values.

We conclude that both the magnitude and the consistent emergence of outlier weights in the same locations across the model are responsible for the emergence of distinct embedding features that BERT heavily relies on.

	1	12	11–12	9–12	7–12	1–12
CE	2.33	2.40	2.67	2.81	2.87	**4.53**
#w	4	4	8	16	24	48

Table 6: BERT-base language modeling cross-entropy loss (*CE*) on the validation set of WikiText corpus, shown by location and number of modified Transformer layers. *#w* denotes the total number of modified weights.

5.2 How Do Outlier Weights Emerge?

In this section, we examine the emergence of outlier dimensions during pre-training. To the best of our knowledge, there are no publicly available BERT pre-training checkpoints available to study these effects. We pre-train[5] a BERT-medium model (chosen due to computational constraints) from scratch on the BookCorpus data (Zhu et al., 2015) and track statistics of the LayerNorm scaling factors and biases. We start from a randomly initialized BERT-medium configuration that has 8 layers with the hidden dimensionality of 512 units. We save checkpoints of the model every 2000 steps, and we track the output LayerNorm weights across all of the model's layers as the training progresses.

Figure 7 shows that both scaling factors and biases begin to diverge from their initialization values quite early (after approximately 50k steps) in the training process. At roughly the same point, both training loss and evaluation perplexity begin to fall off. An interesting question for future work is to clarify whether there is a causal relationship here.

Although the published BERT-medium model had two outlier dimensions, our model had only one dimension for which *both* the scaling factor and the bias exceed their corresponding means by more than three standard deviations.

Due to the gradual emergence of these outliers during pre-training, there is a possibility that thresholding their distance from the rest of the parameters can be used as a litmus test for when the pre-training is complete. We leave the investigation of this hypothesis to future work.

6 Implications and Future Work

The outlier dimension effect we identified may enable **attacks on Transformer-based encoders** that could be used to degrade the model quality while modifying very few weights. Further, since these perturbations to the model do not cause the

[5]We used two RTX-3090 GPUs for pre-training.

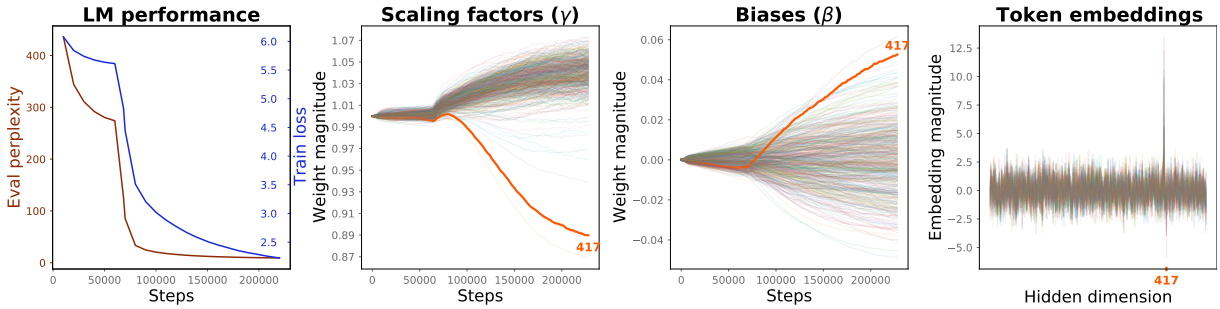

Figure 7: BERT-medium pre-training on the BookCorpus dataset. *(left)* Evaluation perplexity (brown) and train loss (blue) as the training progresses. *(middle)* The changes in the scaling factors and the biases of the output normalization layer. Each line corresponds to one of the 512 dimensions. We highlight (in orange) the `417`-th dimension, for which both the scaling factor and the bias fall out of the three sigma range at the end of pretraining. *(right)* Token embeddings computed for an input sequence that was randomly sampled from the data. Each line corresponds to one input token. The outlier embedding values are marked at the same `417`-th dimension. All the plots are presented for the middle Transformer layer (4).

model to break completely, this may lead to late detection of this attack. To curtail the risks of this attack from affecting deployed Transformer models, we would suggest simple measures such as storing the file checksums for the trained models at a secure location and verifying that the deployed model file matches the checksums.

Another direction for exploiting the phenomenon of outlier dimensions is **pruning**. The studies of model compression using unstructured pruning typically do not consider whether the pruned weights were in the same position throughout the model. Our work suggests that if the outliers were disabled consistently, the drop in performance could be expected to be larger than for random or magnitude-based pruning.

Finally, future work could consider outlier dimensions in the context of **weight initialization**. Our experiments suggest that these dimensions are a normal emergent property of Transformer pretraining. It is possible that higher performance or faster convergence could be achieved by manipulating the initialization to encourage such outliers and experimenting with their number.

7 Conclusion

The main contribution of our work is isolating the phenomenon of a small number of outlier dimensions in Transformer layer outputs which significantly disrupt performance while modifying less than 0.0001% of all parameters of the model. We attribute this phenomenon to an interaction of high-magnitude scaling factors and biases in the same dimension throughout the model, rather than magnitude alone. It emerges early in the training and consistently warps the embedding space.

In case of BERT, the layer output component is LayerNorm. We introduce a method to isolate these outlier dimensions for BERT, and we show that the phenomenon is present in six models of BERT family that we examine. It is also present in four other Transformer-based models (ELECTRA, XLNet, BART, and GPT-2), although the effect of their disabling varies.

8 Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments. This work is funded in part by the NSF award number IIS-1844740 to Anna Rumshisky.

We would like to note that in an independent study concurrent to ours, Luo et al. (2021) report a similar effect of outlier values in the embedding space of BERT and RoBERTa. We invite the reader to look into their study for more details on the outlier features from the perspective of word embeddings.

9 Impact statement

This work analyzes the behavior of layer normalization in the popular BERT family of models, using standard benchmarks. No new models are presented, and no data was collected.

We estimate that experiments presented in this paper consumed 436.8 kWh of energy which resulted in a 108 Kilograms of CO2 emissions computed using the regional average of power generation to carbon emissions.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020. The lottery ticket hypothesis for pre-trained bert networks. In *Advances in Neural Information Processing Systems*, volume 33, pages 15834–15846. Curran Associates, Inc.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

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.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping. *arXiv:2002.06305 [cs]*.

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.

Angela Fan, Edouard Grave, and Armand Joulin. 2019. Reducing Transformer Depth on Demand with Structured Dropout. In *International Conference on Learning Representations*.

Jonathan Frankle and Michael Carbin. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations*.

Prakhar Ganesh, Yao Chen, Xin Lou, Mohammad Ali Khan, Yin Yang, Deming Chen, Marianne Winslett, Hassan Sajjad, and Preslav Nakov. 2020. Compressing large-scale transformer-based models: A case study on BERT. *arXiv preprint arXiv:2002.11985*.

Mitchell Gordon, Kevin Duh, and Nicholas Andrews. 2020. Compressing BERT: Studying the effects of weight pruning on transfer learning. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 143–155, Online. Association for Computational Linguistics.

Demi Guo, Alexander M. Rush, and Yoon Kim. 2020. Parameter-Efficient Transfer Learning with Diff Pruning. *arXiv:2012.07463 [cs]*.

Song Han, Jeff Pool, John Tran, and William J. Dally. 2015. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1135–1143.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1693–1701.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the Dark Secrets of BERT. 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 4356–4365, Hong Kong, China. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ziyang Luo, Artur Kulmizev, and Xiaoxi Mao. 2021. Positional artefacts propagate through masked language model embeddings.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *CoRR*, abs/1609.07843.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are Sixteen Heads Really Better than One? *Advances in Neural Information Processing Systems 32 (NIPS 2019)*.

Toan Q. Nguyen and Julian Salazar. 2019. Transformers without tears: Improving the normalization of self-attention. *CoRR*, abs/1910.05895.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When BERT Plays the Lottery, All Tickets Are Winning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3208–3229, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Sheng Shen, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. 2020. Powernorm: Rethinking batch normalization in transformers. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 8741–8751. PMLR.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Alex Wang, Amapreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. 2020. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR.

Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. 2019. Understanding and improving layer normalization. In *Advances in Neural Information Processing Systems*, volume 32, pages 4381–4391. Curran Associates, Inc.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8BERT: Quantized 8Bit BERT. In *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing (NeurIPS 2019)*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

10 Appendix

10.1 Candidate outlier dimensions of the models.

For each of the BERT-like models we experimented with we present the outlier candidate weights, detected as described in section 3.

Model component	Outliers
output.dense.weight	275, 276, 444
attention.output.dense.bias	193
attention.output.LayerNorm.weight	275, 276, 444
attention.output.LayerNorm.bias	276, 444
output.LayerNorm.weight	121, 262, 444, 276
output.LayerNorm.bias	276, 444

Table 7: BERT-small outlier dimension candidates across model components.

Model component	Outliers
attention.output.dense.bias	92, 400, 476, 17
output.dense.weight	400
output.dense.bias	400
attention.output.LayerNorm.weight	17, 400, 430
attention.output.LayerNorm.bias	192, 400
output.LayerNorm.weight	11, 193, 393, 427, 400
output.LayerNorm.bias	400, 427

Table 8: BERT-medium outlier dimension candidates across model components.

Model component	Outliers
output.dense.weight	308, 381
output.dense.bias	308
attention.output.dense.bias	308
attention.output.LayerNorm.weight	308, 381
attention.output.LayerNorm.bias	145, 308, 381
output.LayerNorm.weight	92, 145, 308, 381, 225
output.LayerNorm.bias	308, 381

Table 9: BERT-base outlier dimension candidates across model components.

10.2 Scaling factor and bias statistics for BERT-base.

For the BERT-base configuration, we present the detailed statistics on per-layer scaling factors and biases of the output LayerNorm (see Table 13). We report per-layer means, standard deviations and counts of the weights falling out of the three sigma range. We also show the values of the outlier weights (`308` and `381`) along with their ranks, where the ranks are computed for the corresponding sorted arrays of weight magnitudes. Note that the outlier weights consistently appear to be *among* the top largest or top smallest LayerNorm weights throughout the model, but are not necessarily the top-1 largest/smallest values.

10.3 Sample language model outputs after disabling outlier LayerNorm weights.

For RoBERTa and BERT, we randomly sample a set of sentences from Wikipedia and BookCorpus, mask multiple input tokens, and use the models for token prediction. We compare the baseline (full) models with the models where select LayerNorm weights are zeroed out across all of the Transformer layers. In particular, we compare the setups where the outlier dimensions (two per model) are disabled as opposed to random dimensions (two per model).

Model component	Outliers
attention.output.dense.bias	757, 327
output.dense.weight	757, 159
output.dense.bias	159, 757
attention.output.LayerNorm.weight	159, 757, 327
attention.output.LayerNorm.bias	159, 757, 327
output.LayerNorm.weight	159, 757, 327
output.LayerNorm.bias	159, 757, 327

Table 10: Multilingual BERT (mBERT) outlier dimension candidates across model components.

Model component	Outliers
output.dense.weight	588
output.dense.bias	588, 494
attention.output.dense.bias	588
attention.output.LayerNorm.bias	77, 217, 453, 551, 588, 496, 731, 494
output.LayerNorm.bias	77, 453, 551, 588, 217, 240, 496, 61, 494

Table 11: Base RoBERTa outlier dimensions across model components.

Model component	Outliers
attention.output.dense.bias	466, 18
output.dense.bias	466, 750, 18, 933
attention.output.LayerNorm.weight	234, 466, 933
attention.output.LayerNorm.bias	9, 71, 136, 234, 327, 706, 466, 474, 929, 933, 18, 143
output.LayerNorm.weight	80, 136, 232, 234, 331, 466, 639, 665, 702, 724, 750, 763, 968, 315, 428, 933, 18, 506, 314
output.LayerNorm.bias	136, 466, 706, 327, 9, 929, 18, 143, 933

Table 12: BERT-large outlier dimension candidates across model components.

	Scaling factors				Biases			
Transf. layer	mean / std	#$> 3\sigma$	308 value / rank	381 value / rank	mean / std	#$> 3\sigma$	308 value / rank	381 value/rank
1	0.756 / 0.056	12	0.343 / 764	0.404 / 762	-0.037 / 0.099	6	-1.325 / 0	0.144 / 78
2	0.870 / 0.069	24	0.400 / 765	0.374 / 766	-0.034 / 0.086	8	-0.678 / 0	0.277 / 5
3	0.851 / 0.052	16	0.408 / 767	0.549 / 765	-0.031 / 0.075	4	-0.070 / 298	0.118 / 103
4	0.811 / 0.044	11	0.562 / 764	0.388 / 767	-0.033 / 0.052	7	0.075 / 174	0.114 / 50
5	0.840 / 0.045	8	0.615 / 763	0.360 / 767	-0.031/ 0.051	8	0.200 / 3	-0.083 / 113
6	0.832 / 0.037	7	0.692 / 763	0.411 / 767	-0.032 / 0.060	6	0.403 / 0	-0.394 / 1
7	0.834 / 0.037	4	0.752 / 752	0.375 / 767	-0.033 / 0.063	5	0.785 / 0	-0.337 / 1
8	0.810 / 0.030	4	1.163 / 0	0.335 / 767	-0.033 / 0.065	2	0.959 / 0	0.304 / 1
9	0.831 / 0.042	6	1.618 / 0	0.262 / 767	-0.035 / 0.062	2	0.129 / 38	0.695/0
10	0.801 / 0.060	7	1.437 / 0	0.254 / 764	-0.032 / 0.057	9	-0.415 / 2	0.258/4
11	0.817 / 0.062	9	1.671 / 0	0.185 / 765	-0.040 / 0.068	5	-0.667 / 1	1.234/0
12	0.633 / 0.027	13	0.273 / 767	0.536 / 758	-0.019 / 0.050	5	0.225 / 0	-0.021/531

Table 13: The statistics of output LayerNorm weights (scaling factors and biases) for all of the Transformer layers of BERT-base.

Input	Ghostbusters was [released] on June 8 , [1984] , to critical [acclaim] and became a cultural phenomenon . It was well [received] for its deft blend of comedy, [action] , and horror , and Murray ' s performance was [repeatedly] singled out for praise .	a filmy coating of [dust] and pebbles had settled onto the block , and [sami] 's hand instinctively jerked forward to swipe the [scratchy] debris off his cheek , then pulled [up] short against the biting [metal] cuffs .	According to the RIAA, the Beatles are the best-[selling] music artists in the United States, with 178 [million] certified units. They have had more number-[one] albums on the [British] charts and sold [more] singles in the UK than any other act.
RoBERTa	Ghostbusters was [released] on June 8 , [1986] , to critical [acclaim] and became a cultural phenomenon . It was well [received] for its deft blend of comedy, [action] , and horror , and Murray ' s performance was [often] singled out for praise .	a filmy coating of [dirt] and pebbles had settled onto the block , and [Sami] 's hand instinctively jerked forward to swipe the [crusty] debris off his cheek , then pulled [up] short against the biting [leather] cuffs .	According to the RIAA, the Beatles are the best-[selling] music artists in the United States, with 178 [million] certified units. They have had more number-[one] albums on the [US] charts and sold [more] singles in the UK than any other act.
Random	Ghostbusters was [released] on June 8 , [1986] , to critical [acclaim] and became a cultural phenomenon . It was well [received] for its deft blend of comedy, [action] , and horror , and Murray ' s performance was [particularly] singled out for praise.	a filmy coating of [dirt] and pebbles had settled onto the block , and [Tsui] 's hand instinctively jerked forward to swipe the [crusty] debris off his cheek , then pulled [up] short against the biting [leather] cuffs .	According to the RIAA, the Beatles are the best-[selling] music artists in the United States, with 178 [million] certified units. They have had more number-[one] albums on the [US] charts and sold [more] singles in the UK than any other act.
Outliers	{ lock was [never] on June 8 , [</s>] , to rely [,] and . It was well [known] for its acker of comedy , [dinner], and horror , and Murray ' s was [ever] , </s> </s>)	a Fre) covering of [humor] and celebcele had </s> </s> </s> </s> , and [</s>i] 's </s> </s> </s> </s> </s> </s> (@ the [brainy] during (@ end) , Then pulled [*] isk ss the wearing [of] cuffs </s>	2017 </s> the RIAA, the Beatles are the [l] music files in the United States, with 178 [Canadian] Certified ols </s> They have had é yl-[million] Deaths on the [Chart] charts and Died [are] Hearts in</s> UK . </s></s></s></s>

Table 14: RoBERTa's masked language model predictions for randomly sampled input sequences. Input masked tokens (blue) are given in brackets. Correctly predicted tokens are shown in green, incorrect but plausible predictions are shown in brown. 48 weights have been modified in total for the *Random* and *Outliers* setups.

Input	he didnt [really] have a plan and he wasnt sure he [could] go through [with] anything , but the [feeling] of doing something was lifting his [spirits] .	ice is water frozen into a [solid] state . [depending] on the presence of impurities such as particles of soil or [bubbles] of air , it can appear [transparent] or a more or less [opaque] bluish - white color .	but the [sound] of the river babbling by the yard and the ducks splashing on the [pond] seemed to be [working] a cure for her [melancholy] .
BERT	he didnt [even] have a plan and he wasnt sure he [could] go through [with] anything , but the [thought] of doing something was lifting his [spirits] .	ice is water frozen into a [frozen] state . [depending] on the presence of impurities such as particles of soil or [particles] of air , it can appear [white] or a more or less [uniform] bluish - white color .	but the [sound] of the river babbling by the yard and the ducks splashing on the [water] seemed to be [providing] a cure for her [fears] .
Random	he didnt [even] have a plan and he wasnt sure he [could] go through [with] anything , but the [thought] of doing something was lifting his [spirits] .	ice is water frozen into a [liquid] state . [depending] on the presence of impurities such as particles of soil or [particles] of air , it can appear [white] or a more or less [uniform] bluish - white color .	but the [sounds] of the river babbling by the yard and the ducks splashing on the [water] seemed to be [just] a cure for her [fears] .
Outliers	he didny [wee] have a plan and he wasnt sure he [would] go through [it] anything , but the [actual] of doing something was lifting his [shoulders] .	that is water turned into a [yu] state . [based] on the presence of impurities such as particles of soil or [breath] of air , it can appear [white] or a more or more [commoning] ing - white color .	but the [sound] of the child babble by the yard and the ducks splashing on the [windows] all to be [in] a replacement for her [ness] .

Table 15: BERT's masked language model predictions for randomly sampled input sequences. Input masked tokens (blue) are given in brackets. Correctly predicted tokens are shown in green, incorrect but plausible predictions are shown in brown. 48 weights have been modified in total for the *Random* and *Outliers* setups.