# On the Analysis and Distillation of Emergent Outlier Properties in Pre-trained Language Models

**Tianyang Zhao[1,2], Kunwar Yashraj Singh[1], Srikar Appalaraju[1], Peng Tang[1]**
**Ying Nian Wu[1], Li Erran Li[1]**
[1]AWS AI Labs, [2]Amazon Alexa AI
{tiaxzhao, sinkunwa, srikara, tangpen, wunyin, lilimam}@amazon.com

## Abstract

A small subset of dimensions within language Transformers' representation spaces emerge as "outliers" during pretraining, encoding critical knowledge sparsely. We extend previous findings on emergent outliers to Encoder-Decoder Transformers and instruction-finetuned models, and tackle the problem of distilling a student Transformer from a larger teacher Transformer. Knowledge distillation reduces model size and cost by transferring knowledge from a larger teacher to a smaller student, necessitating a trade-off among representation dimensions. We show that emergent outlier dimensions contribute significantly more to zero-shot performance than non-outlier dimensions. Based on this, we propose the Emergent Outlier Focused Distillation (EOFD) method, which prioritizes critical outlier dimensions in distillation using a weighted MSE loss. We empirically demonstrate that EOFD outperforms state-of-the-art distillation methods and generalizes well across Encoder-only BERT, Decoder-only GPT-2, and Encoder-Decoder T5 architectures.

## 1 Introduction and Background

Emergent properties in large language models (LLMs) have recently garnered great interest (Wei et al., 2022b; Srivastava et al., 2023; Schaeffer et al., 2023). They have been shown to elicit complex capabilities in LLMs. Emergent properties and features arise spontaneously in these models during self-supervised pretraining, without being explicitly optimized for specialized tasks. Specifically, it has been shown (Kovaleva et al., 2021; Puccetti et al., 2022; Dettmers et al., 2022) that, in Encoder-only BERT family (Devlin et al., 2019) and in Decoder-only GPT family (Radford et al., 2019, 2021; Brown et al., 2020; Zhang et al., 2022) models, a small subset of dimensions within the high-dimensional representation spaces of language Transformers emerge as "outliers" during pretraining: weight or neuron activations with unusually large magnitudes out of several standard deviations from the mean. Interestingly, these emergent outlier features seem to encode critical linguistic knowledge in a sparse way: muting only a few outlier dimensions significantly deteriorates language modeling performance.

In this work, we begin by systematically extending previous findings about emergent outlier properties on pretrained Encoder-only and Decoder-only models to Encoder-Decoder T5 models (Raffel et al., 2020) and to instruction-finetuned Flan-T5 models (Chung et al., 2024) at scale, for the first time. We discover that T5-11B exhibits emergent activation outliers with surprising magnitudes exceeding $10^5$, much larger than the BERT outliers observed by Kovaleva et al. (2021); Puccetti et al. (2022). Furthermore, unlike the rapid emergence of GPT outliers around a model size of 6.7B as found by Dettmers et al. (2022), we notice that the growth in outlier magnitude primarily comes with increasing layer depth rather than model size: from T5-Large to T5-3B and to T5-11B, all with a same number of layers, the outlier magnitudes actually decrease as model size increases. We further find that, agreeing with previous work, outlier dimensions are consistent across layers within either the Encoder or Decoder stack; but, contrary to previous knowledge, the outlier dimensions in the Encoder differ from those in the Decoder.

Moreover, consistent with Dettmers et al. (2022), we notice that outlier features suddenly become crucial to performance when the model size exceeds 6.7B in T5: zeroing out only 4 outlier dimensions out of its 1024 total dimensions (only 192 of its total 11B parameters) in T5-11B degrades absolute performance by 14.7%. However, despite this and Dettmers et al. (2022), for instruction-finetuned Flan-T5, we notice that larger models like Flan-T5-XXL are relatively less sensitive to interventions on outliers than smaller models. Nevertheless, we



*Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8475–8507

confirm that disabling outlier dimensions hinders performance significantly more than disabling the same number of non-outlier dimensions, for all our different settings. For instance, muting as much as 512 non-outlier dimensions in Flan-T5-XXL only drops its performance by 0.64%.

Leveraging our findings on emergent outlier properties, we further tackle the challenging problem of distilling a student language model from a larger and stronger teacher model, by focusing on these critical properties. Training massive models is computationally demanding and requires a vast treasure trove of varied data, both of which are not easily available. Additionally, at inference, hosting such large models also gets progressively expensive. To mitigate this, knowledge distillation (KD) (Hinton et al., 2015) aims at reducing model size without significantly compromising performance by transferring knowledge from a stronger teacher to a smaller student by minimizing the divergence of their soft responses and intermediate features (Gou et al., 2021). For distilling these intermediate features, conventional methods treat different dimensions equally (Jiao et al., 2020a; Fang et al., 2021; Liang et al., 2023a; Wu et al., 2023a).

However, matching student's intermediate features to teacher's is inherently imperfect, as student Transformers typically have fewer dimensions of representation than teachers, necessitating a trade-off among these dimensions. Given our discovery that emergent outlier dimensions contribute much more to performance than non-outlier dimensions, when distilling these intermediate representations, we propose Emergent Outlier Focused Distillation (EOFD). This approach prioritizes these critical outlier dimensions and deprioritizes the less impactful non-outlier dimensions, addressing the dimension trade-off in distillation. Specifically, EOFD computes a weighted MSE loss to weight more on the emergent outlier dimensions, recognized by the standard deviations of neuron activations.

On the standard benchmark of distilling BERT on the 8 tasks and datasets in the General Language Understanding Evaluation (GLUE) (Wang et al., 2019) benchmark, we outperform state-of-the-art distillation methods by a large margin. On the relatively larger datasets in GLUE, student models distilled with EOFD outperform the teacher models. Beyond distilling Encoder-only BERT models, we further demonstrate that EOFD generalizes well to other architectures, including Decoder-only GPT-2 and Encoder-Decoder T5 models. We also provide detailed ablation and analysis. We discuss further related works in Appendix A.

Our contributions are two-fold: (1) For the first time, we extend previous findings on emergent outliers to Encoder-Decoder Transformers and to instruction-finetuned models at scale. We further systematically study their zero-shot performance with interventions on muting representation dimensions by factors of different numbers of disabled outlier/non-outlier dimensions, varying model size, and whether or not the pretrained model is further instruction-finetuned. (2) Leveraging our findings on emergent outlier properties, we propose the Emergent Outlier Focused Distillation (EOFD) method prioritizing these critical outlier dimensions to address the dimension trade-off in knowledge distillation. We empirically show that EOFD outperforms state-of-the-art distillation methods. We further show that EOFD generalizes well across the tasks of distilling Encoder-only BERT, Decoder-only GPT-2, and Encoder-Decoder T5 models.

## 2 A Closer Look at Emergent Outliers in Pre-trained Language Models

In this paper, we use plain lower case letters $x$ for scalars, bold lower case letters $\mathbf{x}$ for vectors, bold upper case letters $\mathbf{X}$ for matrices, and $\mathbf{X}^T$ for transposes. We index each Transformer block/layer in a given Transformer by $l \in \{1, \cdots, L\}$, and each token in token sequence by $i \in \{1, \cdots, N\}$, as illustrated in Fig. 3. Typically, each Transformer block/layer contains a Multi-Head Attention module, a Feed Forward Network (FFN), and some Layer-Norm (LN) transformations. Denote the dimension of a Transformer as $d_{\text{model}}$ as illustrated in Fig. 3; specifically, denote $d_t$ and $d_s$ for the dimensions of a teacher and a student model, respectively. We consider for only one data sample (not batched) for notation simplicity unless otherwise specified.

### 2.1 Recognizing Emergent Outliers by Magnitude in Pre-trained T5 Models

Emergent outliers are the weight entries and the neuron activations emerged in pretrained Transformers which exhibit surprisingly large magnitudes out of several standard deviations (Kovaleva et al., 2021; Puccetti et al., 2022). In Fig. 1 (b-e), we plot some typical histograms of weight/activation magnitudes distributions for some layers in T5-11B model. We refer "weights" as the pretrained parameters, and "activations" as the intermediate

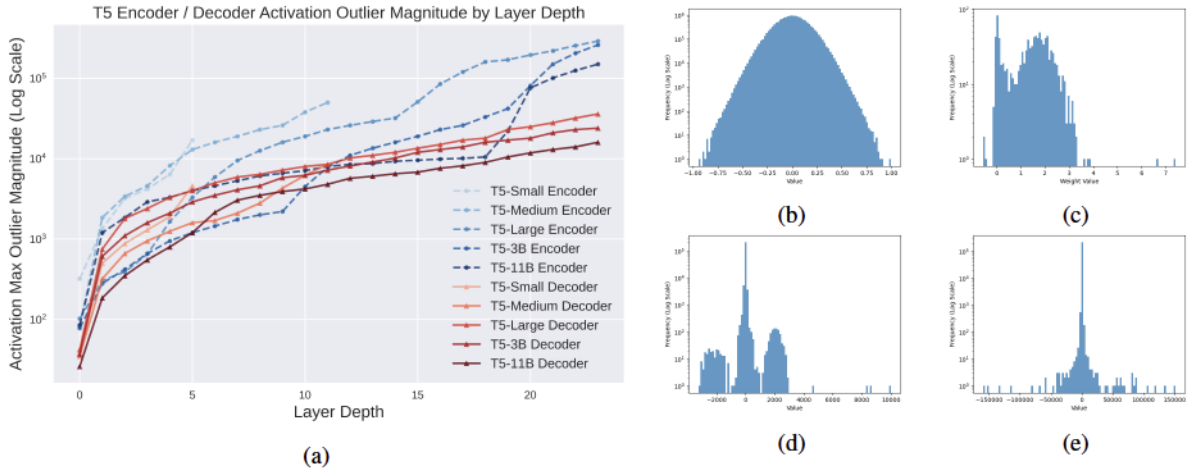



Figure 1: (a) T5 activation outlier magnitude by layer depth and model size; (b) A typical T5-11B weight histogram without outliers; (c) A T5 Layer-Norm weight $\gamma$ histogram with outliers; (d) A T5-11B Encoder hidden state activation histogram with asymmetric outliers and multiple modes; (e) A typical T5-11B Encoder hidden state activation long-tailed histogram with outliers up to a magnitude of 150000. Y-axis is log-scaled. Each of (b-e) is a histogram of weight/activation magnitudes within one layer. More systematic plots are presented in Appendix E.

features / hidden states in pretrained Transformers for some fixed tokenized text input, e.g. as in Appendix F. Specifically, for activations, we investigate the intermediate presentations between each Transformer block, as the purple boxes in Fig. 3: For a given layer $l$, we denote $\mathbf{h}_{l,i} \in \mathbb{R}^{d_{\text{model}}}$ for the intermediate feature vector of the $i$-th token. For each layer $l$, we analyze the activation magnitude distribution of each neuron entry in the intermediate feature vectors for all tokens $\{\mathbf{h}_{l,i}\}_{i=1,\cdots,N}$.

In Fig. 1 (a), we analyze activation outlier scales by model size ranging from T5-Small (60M) to T5-11B, by Encoder/Decoder, and by layer depth (0-th layers are token embeddings). Architecture details of these models are reported in Appendix D. Despite a rapid emergence of GPT-3 outliers at around the model size of 6.7B found by Dettmers et al. (2022), here the plot shows that this growth primarily comes with the increasing layer depth rather than model size: For a given model size, outlier magnitudes increase by layer depth; While from T5-Large to T5-3B and to T5-11B model, the magnitude of outliers actually decreases as model size increases, for both Encoder and Decoder stacks, given that the 3 models all have the same depth of 24+24 layers with different $d_{\text{model}}$.

## 2.2 Emergent Outlier Dimensions are Shared across Layers in T5 and Flan-T5 Models

In the last subsection, we show that outliers with surprisingly large magnitudes emerge in pretrained language models, here we further analyze the patterns of how these outliers are structured throughout the Transformers. For the intermediate feature vectors for all tokens $\{\mathbf{h}_{l,i}\}_{i=1,\cdots,N}$ (purple boxes in Fig. 3) in the $l$-th layer, we refer a "dimension #$j$" to the $j$-th entries $\{h_{l,i,j}\}_{i=1,\cdots,N}$ of these vectors. Previous work found that outlier dimensions are shared across different tokens and layers in Encoder-only BERT models (Kovaleva et al., 2021; Puccetti et al., 2022) and in Decoder-only GPT-3 models larger than 6.7B (Dettmers et al., 2022), so we call a dimension #$j$ either an "outlier dimension #$j$" (green box in Fig. 3) or a "non-outlier dimension #$j$" (orange box in Fig. 3). For weights instead of activations, they found that, for a given layer $l$, the $j$-th weight outlier dimension $\gamma_{l,j}$ in the Layer-Norm directly connected to $\{h_{l,i,j}\}_{i=1,\cdots,N}$ is closely related to the $j$-th activation outlier dimension, as $\gamma_{l,j}$ determines the multiplication scale of that activation dimension:

$$\textbf{BERT: } h_{l,i,j} = \frac{x_{l,i,j} - u_{l,i}}{\sqrt{\sigma_{l,i}^2 + \epsilon}} \cdot \gamma_{l,j} + \beta_{l,j} \quad (1)$$

$$\textbf{T5: } x_{l+1,i,j} = \frac{h_{l,i,j}}{\sqrt{\sigma_{l,i}^2 + \epsilon}} \cdot \gamma_{l,j} \quad (2)$$

where $x_{l,i,j}$ denotes the intermediate feature within the $l$-th Transformer block (blue box in Fig. 3); and for the $l$-th block, $(u_{l,i}, \sigma_{l,i}^2)$ denote the mean and variance of activations for all dimensions within the $i$-th token, $(\gamma_{l,j}, \beta_{l,j})$ denote the learnable scale and bias parameters for each dimension $j$, and $\epsilon$ denotes a small number. Note that BERT is of Post-LN (Post Layer-Norm) style, while T5 and GPT

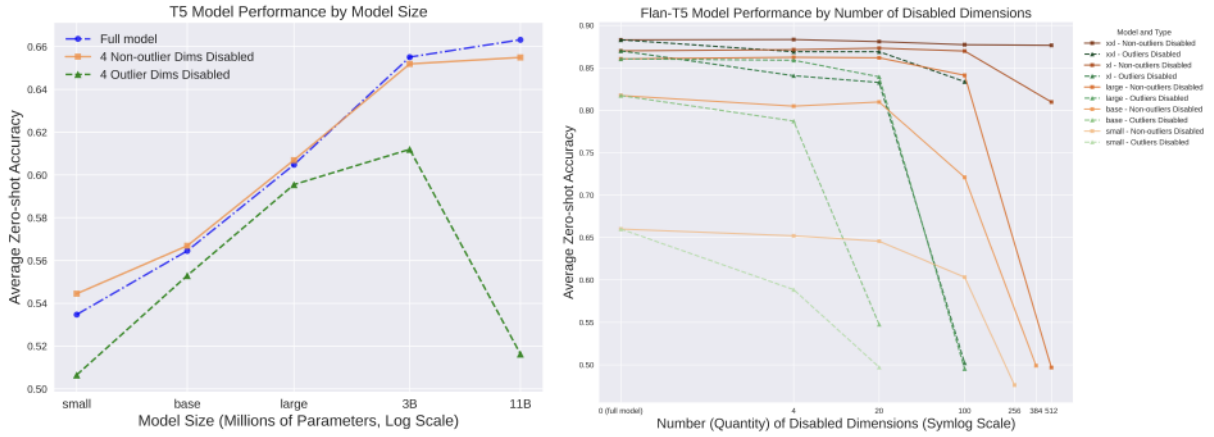



Figure 2: Average zero-shot performance of pretrained T5 and further instruction-finetuned Flan-T5 by model size and by number of disabled outlier/non-outlier dimensions. Full results on each task with standard errors are reported in Table 1 and in Appendix B. X-axis is log-scaled for both subfigures.

models are of Pre-LN style; for their difference, we refer readers to Xiong et al. (2020).

Here, we extend the findings in previous work to Encoder-Decoder T5 and to instruction-finetuned (Ouyang et al., 2022; Wei et al., 2022a; Sanh et al., 2022) Flan-T5 (Chung et al., 2024) models with some correction. Because of the close relationship between Layer-Norm weight outliers and intermediate activation outliers, for each layer $l$, we identify its outlier dimensions $\#j$ where $\gamma_{l,j}$ is out of 3 standard deviations (3-std) of the distribution of $\{\gamma_{l,j}\}_{j=1,\cdots,d_{\text{model}}}$ as in Puccetti et al. (2022). We report the full list of observed outlier dimensions $\#j$ grouped by how many times they are recognized as outlier dimensions across all the $L$ layers for a given Transformer in Appendix H, for T5-Small and Flan-T5-Small to T5-11B and Flan-T5-XXL models. In short, we find that there are some common outlier dimensions within either the Encoder or Decoder stack; but generally, the outlier dimensions in the Encoder stack are different from those dimensions in the Decoder stack. For instance, dim #550 is recognized 23 times as an outlier dimension across all 24 Decoder layers in Flan-T5 XXL; but is never recognized as an outlier dimension in the Encoder layers. This might be caused by how residual connection is implemented within and across the Encoder/Decoder stack. We also find that Decoders tend to have more outlier dimensions than Encoders. Note that, despite (Dettmers et al., 2022), we also observe systemic outlier dimensions in small models: e.g., dim #275 is recognized 5 times as an outlier dimension across all 6 Encoder layers in T5-Small.

### 2.3 Zero-shot Performance of T5 and Flan-T5 Models with Disabled Emergent Outliers

Another defining characteristics for outliers is their surprisingly large contribution to model performance: muting the outlier dimensions, which are only a very small proportion of total dimensions, harm performance seriously (Kovaleva et al., 2021; Dettmers et al., 2022). We study this effect more systematically at scale by factors of different number of disabled outlier/non-outlier dimensions, different model size, and whether or not the pretrained model is further instruction-finetuned.

As supported by previous subsection that outlier dimensions are shared across tokens and layers within the Encoder or within the Decoder stack, we disable common dimensions within Encoder/Decoder stack, but treat Encoder and Decoder independently. To disable dim $\#j$, we zero out the scaling factor for the $j$-th dimension in Layer-Norm for all layers for pretrained T5 and for pretrained and instruction-finetuned Flan-T5 models: $\gamma_{l,j} \leftarrow 0, l \in \{1, \cdots, L\}$; As in Eq. (1), for models in T5 family, this will result in muting dim $\#j$ in activations for all layers and tokens: $x_{l,i,j} \leftarrow 0, l \in \{2, \cdots, L\}, i \in \{1, \cdots, N\}$. In Fig. 2a and in Table 5, we first sort the dimensions decreasingly by how many times they are recognized as outlier dimensions across different layers as in Appendix H, then disable the first 4 outlier dimensions for T5-Small to T5-11B models (green line); As control comparisons, we also randomly

| Model | Configuration | MNLI | QNLI | RTE | SST-2 | Avg (%) |
|---|---|---|---|---|---|---|
| Flan-T5-Small | full model | 0.4243 ± 0.0050 | 0.7403 ± 0.0059 | 0.6029 ± 0.0295 | 0.8727 ± 0.0113 | 66.00 |
| | non-outlier disabled: 4 dims | 0.4092 ± 0.0050 | 0.7219 ± 0.0061 | 0.6065 ± 0.0294 | 0.8704 ± 0.0114 | 65.20 |
| | non-outlier disabled: 20 dims | 0.4254 ± 0.0050 | 0.6720 ± 0.0064 | 0.6173 ± 0.0293 | 0.8681 ± 0.0115 | 64.57 |
| | non-outlier disabled: 100 dims | 0.3749 ± 0.0049 | 0.7095 ± 0.0061 | 0.5054 ± 0.0301 | 0.8234 ± 0.0129 | 60.33 |
| | non-outlier disabled: 256 dims | 0.3428 ± 0.0048 | 0.5102 ± 0.0068 | 0.5054 ± 0.0301 | 0.5459 ± 0.0169 | 47.61 |
| | outlier disabled: 4 dims | 0.3817 ± 0.0049 | 0.6061 ± 0.0066 | 0.5162 ± 0.0301 | 0.8509 ± 0.0121 | 58.87 |
| | outlier disabled: 20 dims | 0.3332 ± 0.0048 | 0.5266 ± 0.0068 | 0.4585 ± 0.0300 | 0.6709 ± 0.0159 | 49.73 |
| | outlier disabled: 100 dims | No enough outliers | | | | - |
| Flan-T5-Base | full model | 0.6674 ± 0.0048 | 0.8774 ± 0.0044 | 0.7870 ± 0.0246 | 0.9232 ± 0.0090 | 81.37 |
| | non-outlier disabled: 4 dims | 0.6556 ± 0.0048 | 0.8772 ± 0.0044 | 0.7653 ± 0.0255 | 0.9209 ± 0.0091 | 80.48 |
| | non-outlier disabled: 20 dims | 0.6806 ± 0.0047 | 0.8741 ± 0.0045 | 0.7653 ± 0.0255 | 0.9186 ± 0.0093 | 80.97 |
| | non-outlier disabled: 100 dims | 0.4724 ± 0.0050 | 0.8336 ± 0.0050 | 0.6787 ± 0.0281 | 0.8991 ± 0.0102 | 72.10 |
| | non-outlier disabled: 384 dims | 0.3285 ± 0.0047 | 0.5135 ± 0.0068 | 0.4946 ± 0.0301 | 0.6594 ± 0.0161 | 49.90 |
| | outlier disabled: 4 dims | 0.6513 ± 0.0048 | 0.7948 ± 0.0055 | 0.7834 ± 0.0248 | 0.9197 ± 0.0092 | 78.73 |
| | outlier disabled: 20 dims | 0.3187 ± 0.0047 | 0.5054 ± 0.0068 | 0.5018 ± 0.0301 | 0.8647 ± 0.0116 | 54.77 |
| | outlier disabled: 100 dims | No enough outliers | | | | - |
| Flan-T5-Large | full model | 0.7238 ± 0.0045 | 0.9043 ± 0.0040 | 0.8737 ± 0.0200 | 0.9404 ± 0.0080 | 86.05 |
| | non-outlier disabled: 4 dims | 0.7292 ± 0.0045 | 0.9050 ± 0.0040 | 0.8773 ± 0.0198 | 0.9381 ± 0.0082 | 86.24 |
| | non-outlier disabled: 20 dims | 0.7323 ± 0.0045 | 0.8960 ± 0.0041 | 0.8773 ± 0.0198 | 0.9415 ± 0.0080 | 86.18 |
| | non-outlier disabled: 100 dims | 0.6930 ± 0.0047 | 0.8720 ± 0.0045 | 0.8664 ± 0.0205 | 0.9335 ± 0.0084 | 84.12 |
| | non-outlier disabled: 512 dims | 0.3579 ± 0.0048 | 0.4946 ± 0.0068 | 0.5235 ± 0.0301 | 0.6112 ± 0.0165 | 49.68 |
| | outlier disabled: 4 dims | 0.7209 ± 0.0045 | 0.9033 ± 0.0040 | 0.8700 ± 0.0202 | 0.9415 ± 0.0080 | 85.89 |
| | outlier disabled: 20 dims | 0.6935 ± 0.0047 | 0.9074 ± 0.0039 | 0.8267 ± 0.0228 | 0.9300 ± 0.0086 | 83.94 |
| | outlier disabled: 100 dims | 0.3290 ± 0.0047 | 0.5312 ± 0.0068 | 0.5415 ± 0.0300 | 0.5803 ± 0.0167 | 49.55 |
| Flan-T5-XL | full model | 0.7279 ± 0.0045 | 0.9422 ± 0.0032 | 0.8628 ± 0.0207 | 0.9472 ± 0.0076 | 87.00 |
| | non-outlier disabled: 4 dims | 0.7284 ± 0.0045 | 0.9422 ± 0.0032 | 0.8664 ± 0.0205 | 0.9484 ± 0.0075 | 87.14 |
| | non-outlier disabled: 20 dims | 0.7404 ± 0.0044 | 0.9411 ± 0.0032 | 0.8628 ± 0.0207 | 0.9484 ± 0.0075 | 87.32 |
| | non-outlier disabled: 100 dims | 0.7313 ± 0.0045 | 0.9378 ± 0.0033 | 0.8628 ± 0.0207 | 0.9472 ± 0.0076 | 86.98 |
| | non-outlier disabled: 512 dims | 0.6220 ± 0.0049 | 0.8935 ± 0.0042 | 0.7978 ± 0.0242 | 0.9255 ± 0.0089 | 80.97 |
| | outlier disabled: 4 dims | 0.6666 ± 0.0048 | 0.9054 ± 0.0040 | 0.8448 ± 0.0218 | 0.9461 ± 0.0077 | 84.07 |
| | outlier disabled: 20 dims | 0.6553 ± 0.0048 | 0.8951 ± 0.0041 | 0.8412 ± 0.0220 | 0.9392 ± 0.0081 | 83.27 |
| | outlier disabled: 100 dims | 0.3307 ± 0.0047 | 0.5059 ± 0.0068 | 0.4982 ± 0.0301 | 0.6743 ± 0.0159 | 50.23 |
| Flan-T5-XXL | full model | 0.7462 ± 0.0044 | 0.9279 ± 0.0035 | 0.8989 ± 0.0181 | 0.9587 ± 0.0067 | 88.29 |
| | non-outlier disabled: 4 dims | 0.7467 ± 0.0044 | 0.9288 ± 0.0035 | 0.8989 ± 0.0181 | 0.9587 ± 0.0067 | 88.33 |
| | non-outlier disabled: 20 dims | 0.7445 ± 0.0044 | 0.9259 ± 0.0035 | 0.8953 ± 0.0184 | 0.9576 ± 0.0068 | 88.08 |
| | non-outlier disabled: 100 dims | 0.7243 ± 0.0045 | 0.9303 ± 0.0034 | 0.8953 ± 0.0184 | 0.9587 ± 0.0067 | 87.72 |
| | non-outlier disabled: 512 dims | 0.7522 ± 0.0044 | 0.9154 ± 0.0038 | 0.8809 ± 0.0195 | 0.9576 ± 0.0068 | 87.65 |
| | outlier disabled: 4 dims | 0.6994 ± 0.0046 | 0.9226 ± 0.0036 | 0.8953 ± 0.0184 | 0.9587 ± 0.0067 | 86.90 |
| | outlier disabled: 20 dims | 0.7030 ± 0.0046 | 0.9248 ± 0.0036 | 0.8881 ± 0.0190 | 0.9599 ± 0.0067 | 86.90 |
| | outlier disabled: 100 dims | 0.6518 ± 0.0048 | 0.9105 ± 0.0039 | 0.8195 ± 0.0232 | 0.9530 ± 0.0072 | 83.37 |

Table 1: Detailed zero-shot performance on each evaluation dataset with standard error of pretrained and instruction-finetuned Flan-T5 models (Chung et al., 2024) by model size and by our interventions with different numbers of disabled outlier/non-outlier dimensions. We sometimes observe even better performance of the models when some of their non-outlier dimensions are disabled than the full models, especially for the larger models.

disable 4 non-outlier dimensions (orange line) instead. In Fig. 2b and in Table 1, we sort and disable the first 4 / 20 / 100 outlier dimensions for Flan-T5-Small to Flan-T5-XXL models, or disabling same or more non-outlier dimensions instead. If there are not enough outliers in either Encoder or Decoder out of 3-std in magnitudes, we only disable those out of 3-std. We evaluate and plot the average zero-shot performance of these models with interventions over several language understanding tasks in Fig. 2. We measure model performance by EleutherAI language model evaluation harness (Gao et al., 2023).

Figure 2a and Table 5 show that, when disabled only 4 outlier dimensions (green), T5 models of all sizes perform significantly worse than full model (blue). Meanwhile, disabling the same number of non-outlier dimensions (orange) do not make significant difference in zero-shot performance for all model sizes. Besides, aligning with Dettmers

et al. (2022), this performance degrading phenomena emerges especially for very large models: disabling only 4 out of 1024 total dimensions (only 192 of its total 11B parameters) in T5-11B results in a 14.7% absolute performance drop from the full model. Figure 2b and Table 1 further show that, muting outlier dimensions (green) hinder the performance of Flan-T5 models significantly more than disabling the same number of non-outlier dimensions (orange), for all model sizes and for different numbers of disabling dimensions. Note that, in contrast to the findings on T5 models and despite Dettmers et al. (2022), for instruction-finetuned Flan-T5 models, larger models like Flan-T5-XXL are relatively less sensitive to outliers than smaller models. Furthermore, Flan-T5-XXL is very robust to non-outliers: disabling 512 non-outlier dimensions in Flan-T5-XXL only drops its performance by 0.64%.

## 3 Distilling Transformers: Background

For distilling Transformers, $\mathcal{L}_{\text{PRED}}$, $\mathcal{L}_{\text{ATT}}$, and $\mathcal{L}_{\text{HID}}$ are commonly applied to transfer knowledge from the teacher $t$ to the student $s$. The prediction logits distillation loss $\mathcal{L}_{\text{PRED}}$ minimizes the divergence between the soft response from $t$ and $s$: $\mathcal{L}_{\text{PRED}} = \text{CE}(\mathbf{z}^s/\tau_d, \mathbf{z}^t/\tau_d)$, where $\tau_d$ denotes the temperature, $\mathbf{z}^s$ and $\mathbf{z}^t$ refer to the classification logits (commonly over the tokens in vocabulary) from $s$ and $t$, and CE denotes Cross Entropy, i.e. $p_j = \frac{\exp(z_j/\tau_d)}{\sum_k \exp(z_k/\tau_d)}$ and $\mathcal{L}_{\text{PRED}} = \sum_j p_j^t \cdot \log(p_j^s)$. The attention map distillation loss $\mathcal{L}_{\text{ATT}}$ minimizes the average Mean Squared Error (MSE) between the attention matrices of each head of $t$ and $s$: $\mathcal{L}_{\text{ATT}} = \frac{1}{H}\sum_{h=1}^{H} \text{MSE}(\mathbf{A}_h^s, \mathbf{A}_h^t)$ (Jiao et al., 2020a), where $H$ denotes the number of attention heads, $\mathbf{A}_h \in \mathbb{R}^{N\times N}$ denotes the attention map of the $h$-th head, and $N$ refers to the sequence length of tokens. The intermediate representation distillation loss $\mathcal{L}_{\text{HID}}$ minimizes the divergence between the hidden state matrices for each Transformer block, as in Eq. (3), and we will investigate this loss by outlier or non-outlier dimensions.

## 4 Emergent Outlier Focused Distillation

We have just shown that different representation dimensions in pretrained language models do not contribute equally to performance: Muting the outlier dimensions harm performance significantly, while disabling as much as 512 non-outlier dimensions do not for Flan-T5-XXL. Therefore, when distilling intermediate representations, we propose to focus on these more important outlier dimensions and pay relatively less attention to the non-outlier dimensions.

For notation clarity, we consider for a single data sample (not batched) unless specified otherwise. As in Fig. 3, for the $l$-th layer, denote $\mathbf{H}_l^t \in \mathbb{R}^{N\times d_t}$ for the teacher's representations comprised of $\{(\mathbf{h}_{l,i}^t)^T\}_{i=1,\cdots,N}$, and denote $\mathbf{H}_l^s \in \mathbb{R}^{N\times d_s}$ for the student's. Conventional intermediate representation distillation loss computes the Mean Squared Error (MSE) as below, where $\mathbf{W}^{\text{proj}} \in \mathbb{R}^{d_s\times d_t}$ is a learnable projection from the student's hidden space to the teacher's:

$$\mathcal{L}_{\text{HID}} = \sum_{l=1}^{L} \text{MSE}(\mathbf{H}_l^s\mathbf{W}^{\text{proj}}, \mathbf{H}_l^t) \tag{3}$$

We propose to compute a weighted MSE loss, the Emergent Outlier Focused Distillation loss $\mathcal{L}_{\text{EOFD}}$, instead, weighting more on the outlier dimensions and weighting less on other dimensions, recognized by the standard deviations of activations of these dimensions in the teacher:

$$\mathcal{L}_{\text{EOFD}} = \frac{1}{Nd_t}\sum_{l=1}^{L}\sum_{j=1}^{d_t}\left(w_{l,j}^{\text{eofd}}\sum_{i=1}^{N}\left((\mathbf{H}_l^s\mathbf{W}^{\text{proj}})_{ij} - (\mathbf{H}_l^t)_{ij}\right)^2\right) \tag{4}$$

where $(\mathbf{X})_{ij} \in \mathbb{R}$ denotes the entry of the $i$-th row and the $j$-th column in a matrix $\mathbf{X}$; and for the $l$-th layer, for each teacher hidden dimension $\#j \in \{1, \cdots, d_t\}$, its emergent outlier focused distillation weight $w_{l,j}^{\text{eofd}} \in \mathbb{R}$ is determined by the activation standard deviations (std) of that dimension. Dimensions with larger std are assigned with larger weights and vice versa. Note that as in Eq. (1), a dimension $\#j$ with a large std in activations is closely related to a large scale factor of that dimension $\#j$ in Layer-Norm: $\gamma_{l,j}$; and we have shown in the previous section that these kind of dimensions contribute more to model performance. Formally, we compute $w_{l,j}^{\text{eofd}}$ as follows:

For a given layer $l$, we first compute the activation standard deviation $\sigma_{l,j}$ for each teacher hidden dimension $\#j$. $\sigma_{l,j}$ is computed across all the tokens in the sequence and across all the data instances in the mini-batch: $\sigma_{l,j} = \sigma(\{({}^b\mathbf{H}_l^t)_{i,j} | i \in \{1, \cdots, N\}, b \in \{1, \cdots, \text{batch size}\}\})$.

To compute $w_{l,j}^{\text{eofd}}$, we normalize $\sigma_{l,j}$ by dividing it by the mean of these standard deviations for the given layer $l$, so that the normalized $\sigma_{l,j}$ for a given layer $l$ has a mean of 1. We then raise its quotient

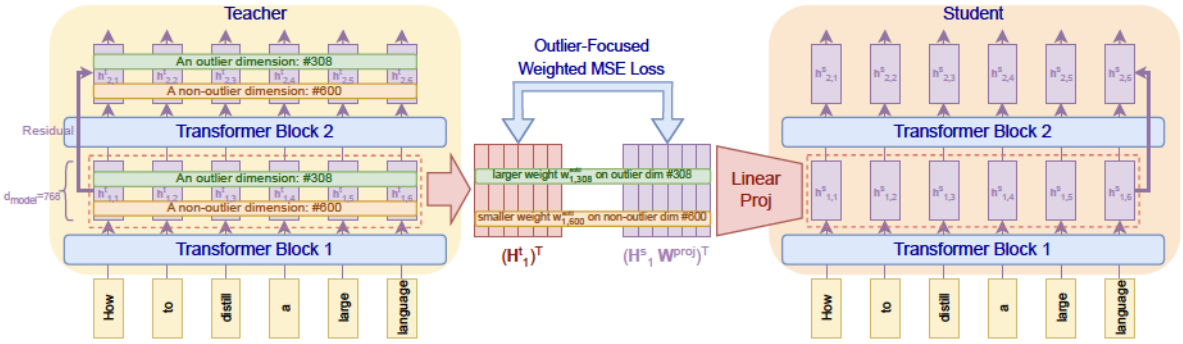



Figure 3: An illustration of the proposed Emergent Outlier Focused Distillation method.

to the power of $p$ to tune the strength of weighting. We use $p \in \{0.5, 1\}$ for our experiments. Note that when $p = 0$, with uniform weighting of $w^{\text{eofd}}_{l,j} \equiv 1$, the EOFD loss in Eq. (4) will be degraded to the vanilla MSE loss in Eq. (3):

$$w^{\text{eofd}}_{l,j} = \left( \frac{\sigma_{l,j}}{\sum_{k=1}^{d_t} \sigma_{l,k}/d_t} \right)^p \tag{5}$$

We provide PyTorch (Paszke et al., 2019) code for the proposed EOFD loss in Appendix I.

## 5 Experiments on EOFD

We distill BERT (Devlin et al., 2019), GPT-2 (Radford et al., 2019) and T5 (Radford et al., 2019) models and evaluate on the General Language Understanding Evaluation (GLUE) (Wang et al., 2019) benchmark. We report model architectures and dataset details in Appendix D.

### 5.1 Distilling BERT on the GLUE benchmark

For fair comparison with state-of-the-art knowledge distillation methods, we first distill BERT-base (Devlin et al., 2019) to a 6-layer small BERT model EOFD-BERT$_6$ and a 4-layer tiny BERT model EOFD-BERT$_4$ on the GLUE benchmark. We build our code upon TinyBERT (Jiao et al., 2020a) [1]. We initiate our student models with their pretrained parameters and conduct finetuning distillation with our proposed EOFD method. In finetuning distillation, we adopt the same data pre-processing and two-step distillation pipeline as in TinyBERT. As them, in the first step, we distill the intermediate representations by applying the attention-map distillation loss $\mathcal{L}_{\text{ATT}}$ and our proposed loss $\mathcal{L}_{\text{EOFD}}$ on all student layers (all purple boxes in Fig. 3). Note that BERT-base has 12 layers, so as Jiao et al. (2020a), we distill the $\#\{2, 4, 6, 8, 10, 12\}$ layers in the teacher to the 6 layers in EOFD-BERT$_6$ respectively, and distill the $\#\{3, 6, 9, 12\}$ layers to the 4 layers in EOFD-BERT$_4$ respectively. We also distill the token embedding layer, as in Jiao et al. (2020a), with $\mathcal{L}_{\text{EOFD}}$. In the second step, we distill the prediction logits by applying $\mathcal{L}_{\text{PRED}}$ to match the final output logits between teachers and students. As in Liang et al. (2023a), on the tiny datasets of RTE, MRPC, and STS-B, we initiate our models from our MNLI step-1 finetuned models. We report further details for training our teacher and student models, e.g. hyper-parameters, seeds, hardware platforms, etc. in Appendix G. We compare with baseline models and recent state-of-the-art knowledge distillation methods of the two student size settings in Table 2. We report the performance of Sanh et al. (2019), Wang et al. (2020a), Wang et al. (2021), Liang et al. (2023a) as was reported in Liang et al. (2023a). Our distilled models outperform state-of-the-art performance by a large margin in average score and on most of the datasets individually for both student size settings.

### 5.2 Analysis and Ablation on BERT

We first analyze our outlier-focused distillation weights $w^{\text{eofd}}_{l,j}$ in Eq. (5) for the BERT distillation experiments in the previous subsection. Take the final layer $l = 6$ for instance, as expected, the proposed method focuses more on distilling the outlier dimensions: dim #308, #381, #251, #539 are assigned with the largest weights $w^{\text{eofd}}_{6,j}$ of $8.42, 1.61, 1.58, 1.58$, respectively; due to the largest standard deviations $\sigma_{6,j}$ of activation magnitudes of these dimensions: $5.04, 0.96, 0.95, 0.94$, compared with the medium std of all dimensions around $0.58$. This also aligns with the reported BERT-base outlier dimensions of #308 and #381 in Kovaleva et al. (2021); Puccetti et al. (2022).

[1] https://github.com/huawei-noah/Pretrained-Language-Model/tree/master/TinyBERT

| BERT Models (Devlin et al., 2019) | Params (M) | MNLI Acc (m/mm) | QQP Acc/F1 | QNLI Acc | SST-2 Acc | CoLA Acc | RTE Acc | MRPC Acc/F1 | STS-B P/S | Avg Score |
|---|---|---|---|---|---|---|---|---|---|---|
| BERT-base (teacher) | 109 | 84.6/85.1 | 91.3/88.2 | 91.8 | 93.2 | 59.1 | 81.6 | 89.2/92.3 | 89.3/89.0 | 85.0 |
| DistilBERT$_6$ (Sanh et al., 2019) | 66 | 82.4/82.5 | 90.4/87.1 | 89.2 | 90.9 | 53.5 | 75.5 | 86.5/90.5 | 87.9/87.8 | 82.1 |
| TinyBERT$_6$-GD (Jiao et al., 2020a) | 66 | 83.5/- | 90.6/- | 90.5 | 91.6 | 42.8 | 77.3 | 88.5/91.6 | 89.0/88.9 | 81.9 |
| TinyBERT$_6$-GD+TD (Jiao et al., 2020a) | 66 | 84.5/84.5 | 91.1/88.0 | 91.1 | **93.0** | 54.0 | 73.4 | 86.3/90.6 | **90.1**/89.6 | 83.0 |
| MiniLM$_6$ (Wang et al., 2020a) | 66 | 84.0/- | 91.0/- | 91.0 | 92.0 | 49.2 | - | -/- | -/- | - |
| MiniLMv2$_6$ (Wang et al., 2021) | 66 | 84.0/- | 91.1/- | 90.8 | 92.4 | 52.5 | 78.0 | 88.7/92.0 | 89.3/89.2 | 83.6 |
| HomoBERT-base (Liang et al., 2023a) [ICLR] | 65 | 84.2/84.3 | 91.2/87.9 | 90.7 | 92.7 | 55.9 | 77.6 | **89.0**/91.9 | 89.5/89.2 | 83.8 |
| TED-BERT$_6$ (Liang et al., 2023b) [ICML] | 66 | 83.4/84.0 | -/- | - | 91.7 | - | 68.8 | -/- | -/- | - |
| SKDBERT$_6$ (Ding et al., 2023a) [AAAI] | 66 | 84.1/83.7 | 91.0/87.9 | 91.4 | 92.9 | - | 75.5 | **89.0/92.1** | 89.2/88.7 | - |
| AD-KD (Wu et al., 2023b) [ACL] | 66 | 83.4/84.2 | 91.2/- | 91.2 | 91.9 | 58.3 | 70.9 | -/91.2 | 89.2/- | 83.5 |
| WID (Wu et al., 2024b) [NAACL] | 55 | 82.9/- | 91.0/- | 90.1 | 92.4 | **61.7** | 70.4 | 88.2/- | 87.9/- | - |
| **Ours EOFD-BERT$_6$** | 66 | **84.9/85.2** | **91.5/88.7** | **91.7** | 92.3 | 56.3 | **80.1** | 88.5/91.9 | 89.9/ **89.7** | **84.4** |
| BERT-small (Devlin et al., 2019) | 28.6 | 78.8/78.9 | 89.9/86.5 | 87.0 | 88.2 | 36.1 | 70.8 | 85.8/90.1 | 87.7/87.7 | 78.1 |
| MiniLM$_3$ (Wang et al., 2020a) | 17.0 | 78.8/- | 88.8/85.0 | 84.7 | 89.3 | 15.8 | 66.4 | 81.9/88.2 | 85.4/85.5 | 74.1 |
| TinyBERT$_4$-GD (Jiao et al., 2020a) | 14.5 | 80.4/80.9 | 88.7/85.3 | 85.7 | 89.7 | 18.6 | 71.1 | 84.6/89.1 | 87.0/87.2 | 75.8 |
| TinyBERT$_4$-GD+TD (Jiao et al., 2020a) | 14.5 | **82.8**/82.9 | -/- | - | - | **50.8** | - | 85.8/- | -/- | - |
| HomoBERT-tiny (Liang et al., 2023a) [ICLR] | 14.1 | 81.2/81.3 | 89.9/86.6 | 87.8 | 90.1 | 37.0 | 70.8 | 87.3/90.7 | 87.6/87.5 | 79.0 |
| HomoBERT-xsmall (Liang et al., 2023a) [ICLR] | 15.6 | 81.5/81.8 | 90.0/86.7 | 88.0 | 90.3 | 40.8 | 71.5 | 87.7/91.0 | 88.3/88.0 | 79.8 |
| **Ours EOFD-BERT$_4$** | 14.5 | 82.6/**83.1** | **90.6/87.5** | **89.3** | **92.2** | 38.2 | **76.9** | **88.4/ 91.9** | **88.4/ 88.2** | **80.9** |

Table 2: Distillation benchmark performance of BERT models on GLUE test-dev set.

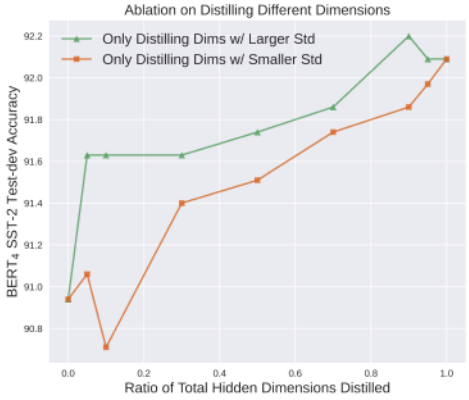



Figure 4: Ablation on BERT$_4$ with $d_{\text{model}} = 312$, only distilling dimensions with largest or smallest $\sigma_{l,j}$.

We also visualize the representation activation histograms of our distilled EOFD-BERT$_6$ model on the MNLI dataset, in comparison with that of the TinyBERT (Jiao et al., 2020a) distilled BERT$_6$ model in Appendix C. The activation histogram of our distilled model exhibits a more long-tailed distribution with some outliers.

Another question is that, if the outlier dimensions contribute much more to performance than non-outlier dimensions, and if weighting the outliers more while distilling can boost performance, then how will it perform if we only distill the outlier dimensions or only distill the non-outlier dimensions? To answer this, we conduct another ablation in Fig. 4. In the green line, for each layer $l$, we only distill the representation dimensions $\{j|\sigma_{l,j} > \alpha\}$ with the largest $x(\%)$ standard deviation $\sigma_{l,j}$ among all dimensions; while in the orange line, we only distill the dimensions $\{j|\sigma_{l,j} < \beta\}$ with the smallest $x(\%)$ std. The X-axis represents the threshold ratio $x$. When $x = 0.0$ we do not distill any dimension, and when $x = 1.0$ we distill all dimensions. No weighting is applied in the MSE loss, so each dimension is treated equally. We always apply $\mathcal{L}_{\text{PRED}}$ and do not apply $\mathcal{L}_{\text{ATT}}$ in this ablation. We ablate on the SST-2 (Socher et al., 2013) dataset, keep other settings the same as in the previous subsection, and report the test-dev accuracy. Fig. 4 shows that ($x = 0.05$, green): only distilling 5% of the dimensions with the largest standard deviations (the outlier dimensions) can recover most of the distilling performance; while distilling the same number of non-outlier dimensions ($x = 0.05$, orange) do not boost performance. Besides, distilling dimensions with larger std (green) consistently outperforms distilling a same number of dimensions with smaller std (orange).

### 5.3 Generalization and Ablation on Distilling GPT-2 and T5 Models

To evaluate how our proposed method generalizes to other model families and for further ablation, we also distill the Decoder-only GPT-2-medium (Radford et al., 2019) to GPT-2 and distill the Encoder-Decoder T5-base (Raffel et al., 2020) to T5-small on several datasets in the GLUE benchmark. We ablate on the effect of ground-truth (GT) supervision without distillation, prediction logits distillation $\mathcal{L}_{\text{PRED}}$, conventional hidden state distillation $\mathcal{L}_{\text{HID}}$, our proposed $\mathcal{L}_{\text{EOFD}}$, and the effect of proxy

| GPT-2 (Radford et al., 2019) | Params (M) | GT | $\mathcal{L}_{\text{PRED}}$ | $\mathcal{L}_{\text{HID}}$ | $\mathcal{L}_{\text{EOFD}}$ (Ours) | LN-KD (Ours) | PT-KD (Ours) | MNLI Acc m | MNLI Acc mm | QQP Acc | QQP F1 | QNLI Acc | CoLA Acc | Avg Score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| GPT-2-medium (teacher) | 345 | ✓ | | | | | | 84.6 | 85.3 | 91.0 | 88.1 | 90.4 | 52.9 | 79.4 |
| GPT-2 | 117 | ✓ | | | | | | 81.6 | 82.4 | 89.7 | 86.3 | 87.4 | 41.9 | 74.8 |
| GPT-2 reported by Li et al. (2021) | 117 | ✓ | | | | | | 82.3 | - | 89.5 | - | 88.6 | 43.2 | - |
| Distilled GPT-2 | 117 | | ✓ | | | | | 81.8 | 82.9 | 90.0 | 86.8 | 89.0 | 43.7 | 75.9 |
| Distilled GPT-2 | 117 | | ✓ | ✓ | | | | 82.4 | 82.9 | 90.3 | 87.1 | 88.9 | 43.7 | 76.0 |
| Distilled GPT-2 w/ our EOFD | 117 | | ✓ | | ✓ | | | 83.0 | 83.1 | 90.4 | 87.2 | 88.8 | 44.2 | 76.2 |
| Distilled GPT-2 w/ our EOFD | 117 | | ✓ | | ✓ | ✓ | | 83.2 | 83.6 | **90.5** | **87.3** | **89.4** | 44.7 | 76.6 |
| Distilled GPT-2 w/ our EOFD | 117 | | ✓ | | ✓ | ✓ | ✓ | **83.4** | **83.8** | **90.5** | **87.3** | 89.1 | **47.4** | **77.3** |

Table 3: Ablation performance of distilling GPT-2 (Radford et al., 2019) on GLUE test-dev set.

| T5 Models (Radford et al., 2019) | Params (M) | GT | $\mathcal{L}_{\text{PRED}}$ | $\mathcal{L}_{\text{HID}}$ | $\mathcal{L}_{\text{EOFD}}$ (Ours) | MNLI Acc m | MNLI Acc mm | QQP Acc | QQP F1 | QNLI Acc | SST-2 Acc | CoLA Acc | Avg Score |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| T5-base reported by Raffel et al. (2020) | 220 | ✓ | | | | 87.1 | 86.2 | 89.4 | 72.6 | 93.7 | 95.2 | 51.1 | 82.2 |
| T5-base reproduced (teacher) | 220 | ✓ | | | | 86.8 | 87.1 | 91.8 | 89.0 | 92.8 | 94.7 | 58.0 | 85.7 |
| T5-small reported by Raffel et al. (2020) | 60 | ✓ | | | | 82.4 | 82.3 | 88.0 | 70.0 | 90.3 | 91.8 | 41.0 | 78.0 |
| T5-small reproduced | 60 | ✓ | | | | 82.2 | 82.9 | 89.4 | 85.7 | 89.1 | 91.4 | 39.5 | 80.0 |
| Distilled T5-small | 60 | | ✓ | | | 82.9 | 83.6 | 90.1 | 86.5 | 89.8 | 91.5 | 42.2 | 80.9 |
| Distilled T5-small | 60 | | ✓ | ✓ | | 83.4 | 83.7 | 90.7 | 87.4 | 90.1 | 91.9 | 42.6 | 81.4 |
| Distilled T5-small w/ our EOFD | 60 | | ✓ | | ✓ | **83.5** | **84.4** | **90.8** | **87.6** | **90.5** | **92.5** | **43.1** | **81.8** |

Table 4: Ablation performance of distilling T5 (Raffel et al., 2020) on GLUE test-dev set.

pretraining distillation (PT-KD) with EOFD, in Table 3 and Table 4. For proxy pretraining, we initiate from the Hugging Face (HF) pretrained GPT-2 model and continue pretrain it with causal language modeling logits distillation loss and the proposed EOFD loss on the HF BookCorpus dataset (Zhu et al., 2015) (3GB) for 3 epochs. We build our code upon the HF Transformers repository [2] (Wolf et al., 2020). We report training details in Appendix G.

Note that unlike BERT models, T5 and GPT-2 are pre-Layer-Norm (pre-LN) Transformers (Xiong et al., 2020): their intermediate representations $\mathbf{H}_l$ (purple boxes in Fig. 3) between Transformer blocks are not normalized. Hence, as we analyzed in Section 2.1, they may contain activation outliers of very large magnitude to the scale over $10^4$, as shown in Fig. 1a; and the scale between the teacher features $\mathbf{H}_l^t$ and the student features $\mathbf{H}_l^s$ may differ significantly. Therefore, specially for computing the distilling loss, in some ablative settings, for each layer $l$, we propose to apply an additional learnable Layer-Norm (LN-KD) on $\mathbf{H}_l^s$ and an additional frozen Layer-Norm on $\mathbf{H}_l^t$, to match the scale difference before applying EOFD:

$$\tilde{h}^s_{l,i,j} = \frac{h^s_{l,i,j} - u^s_{l,i}}{\sqrt{(\sigma^s_{l,i})^2 + \epsilon}} \cdot \gamma_{l,j} + \beta_{l,j}; \quad \tilde{h}^t_{l,i,j} = \frac{h^t_{l,i,j} - u^t_{l,i}}{\sqrt{(\sigma^t_{l,i})^2 + \epsilon}} \tag{6}$$

[2] https://github.com/huggingface/transformers/tree/main/examples/pytorch/text-classification

## 6 Conclusion

We have analyzed the emergent outlier phenomenon and its effect on performance for pretrained T5 and instruction-finetuned Flan-T5 models of size varying from 60M to 11B. Based on the analysis, we have proposed, for the first time, to leverage these findings on outliers for more effective knowledge distillation methods, and have empirically shown that our proposed EOFD method achieves SOTA performance.

## 7 Limitations

Due to constraint of computational resource, our distillation experiments are limited within 345M parameters. Apart from the emergent outlier phenomena, recent researches have also found spontaneously emerged sparsity Li et al. (2022b), token-specific large magnitude activation (Sun et al., 2024), and low-rank memory storage/editing mechanisms in FFN (Meng et al., 2022; Sharma et al., 2024) for Transformers. It still remains an open question whether these phenomena are related; and the dynamics of how outliers are formed during pretraining is still unveiled theoretically. We hope to address these limitations of our current research in future work.

## References


Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations*.

AI@Meta. 2024. Llama 3 model card.

Alex Andonian, Shixing Chen, and Raffay Hamid. 2022. Robust cross-modal representation learning with progressive self-distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16430–16441.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report. *Preprint*, arXiv:2305.10403.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth PASCAL recognizing textual entailment challenge.

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.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Xianing Chen, Qiong Cao, Yujie Zhong, Jing Zhang, Shenghua Gao, and Dacheng Tao. 2022. Dearkd: Data-efficient early knowledge distillation for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12052–12062.

Zihang Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. 2017. Quora question pairs.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Gpt3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, pages 30318–30332. Curran Associates, Inc.

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.

Zixiang Ding, Guoqing Jiang, Shuai Zhang, Lin Guo, and Wei Lin. 2023a. Skdbert: Compressing bert via stochastic knowledge distillation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6):7414–7422.

Zixiang Ding, Guoqing Jiang, Shuai Zhang, Lin Guo, and Wei Lin. 2023b. Skdbert: Compressing bert via stochastic knowledge distillation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(6):7414–7422.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Zhiyuan Fang, Jianfeng Wang, Xiaowei Hu, Lijuan Wang, Yezhou Yang, and Zicheng Liu. 2021. Compressing visual-linguistic model via knowledge distillation. *ICCV*.

Prakhar Ganesh, Yao Chen, Xin Lou, Mohammad Ali Khan, Yin Yang, Hassan Sajjad, Preslav Nakov, Deming Chen, and Marianne Winslett. 2021. Compressing large-scale transformer-based models: A case study on bert. *Transactions of the Association for Computational Linguistics*, 9:1061–1080.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. 2021. Open-vocabulary object detection via vision and language knowledge distillation. In *ICLR*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*.

Ruifei He, Shuyang Sun, Jihan Yang, Song Bai, and Xiaojuan Qi. 2022. Knowledge distillation as efficient pre-training: Faster convergence, higher data-efficiency, and better transferability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9161–9171.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.

Ananya Harsh Jha, Tom Sherborne, Evan Pete Walsh, Dirk Groeneveld, Emma Strubell, and Iz Beltagy. 2023. How to train your (compressed) large language model. *Preprint*, arXiv:2305.14864.

Yuxin Jiang, Chunkit Chan, Mingyang Chen, and Wei Wang. 2023. Lion: Adversarial distillation of proprietary large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3134–3154, Singapore. Association for Computational Linguistics.

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

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020b. 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.

Junmo Kang, Wei Xu, and Alan Ritter. 2023. Distill or annotate? cost-efficient fine-tuning of compact models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11100–11119, Toronto, Canada. Association for Computational Linguistics.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, Luke Zettlemoyer, and Pre-training Bart. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *ACL*.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2665–2679, Toronto, Canada. Association for Computational Linguistics.

Tianda Li, Yassir El Mesbahi, Ivan Kobyzev, Ahmad Rashid, Atif Mahmud, Nithin Anchuri, Habib Hajimolahoseini, Yang Liu, and Mehdi Rezagholizadeh. 2021. A short study on compressing decoder-based language models. *ArXiv*, abs/2110.08460.

Zheng Li, Zijian Wang, Ming Tan, Ramesh Nallapati, Parminder Bhatia, Andrew Arnold, Bing Xiang, and Dan Roth. 2022a. Dq-bart: Efficient sequence-to-sequence model via joint distillation and quantization. *arXiv preprint arXiv:2203.11239*.

Zonglin Li, Chong You, Srinadh Bhojanapalli, Daliang Li, Ankit Singh Rawat, Sashank J. Reddi, Kenneth Q Ye, Felix Chern, Felix X. Yu, Ruiqi Guo, and Surinder Kumar. 2022b. The lazy neuron phenomenon: On emergence of activation sparsity in transformers. In *International Conference on Learning Representations*.

Chen Liang, Haoming Jiang, Zheng Li, Xianfeng Tang, Bing Yin, and Tuo Zhao. 2023a. Homodistil: Homotopic task-agnostic distillation of pre-trained transformers. In *The Eleventh International Conference on Learning Representations*.

Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2023b. Less is more: Task-aware layer-wise distillation for language model compression. In *International Conference on Machine Learning*, pages 20852–20867. PMLR.

Kevin J Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. 2021. Mixkd: Towards efficient distillation of large-scale language models. *Preprint*, arXiv:2011.00593.

Zongyang Ma, Guan Luo, Jin Gao, Liang Li, Yuxin Chen, Shaoru Wang, Congxuan Zhang, and Weiming Hu. 2022. Open-vocabulary one-stage detection with hierarchical visual-language knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14074–14083.

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching small language models to reason. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1773–1781, Toronto, Canada. Association for Computational Linguistics.

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

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever,

Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. *Preprint*, arXiv:1912.01703.

Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. 2019. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5007–5016.

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.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023. Are emergent abilities of large language models a mirage? In *Thirty-seventh Conference on Neural Information Processing Systems*.

Pratyusha Sharma, Jordan T. Ash, and Dipendra Misra. 2024. The truth is in there: Improving reasoning in language models with layer-selective rank reduction. In *The Twelfth International Conference on Learning Representations*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ambrose Slone, Ameet Rahane, Anantharaman S. Iyer, Anders Johan Andreassen, Andrea Madotto, Andrea Santilli, Andreas Stuhlmüller,

Andrew M. Dai, Andrew La, Andrew Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakaş, B. Ryan Roberts, Bao Sheng Loe, Barret Zoph, Bartłomiej Bojanowski, Batuhan Özyurt, Behnam Hedayatnia, Behnam Neyshabur, Benjamin Inden, Benno Stein, Berk Ekmekci, Bill Yuchen Lin, Blake Howald, Bryan Orinion, Cameron Diao, Cameron Dour, Catherine Stinson, Cedrick Argueta, Cesar Ferri, Chandan Singh, Charles Rathkopf, Chenlin Meng, Chitta Baral, Chiyu Wu, Chris Callison-Burch, Christopher Waites, Christian Voigt, Christopher D Manning, Christopher Potts, Cindy Ramirez, Clara E. Rivera, Clemencia Siro, Colin Raffel, Courtney Ashcraft, Cristina Garbacea, Damien Sileo, Dan Garrette, Dan Hendrycks, Dan Kilman, Dan Roth, C. Daniel Freeman, Daniel Khashabi, Daniel Levy, Daniel Moseguí González, Danielle Perszyk, Danny Hernandez, Danqi Chen, Daphne Ippolito, Dar Gilboa, David Dohan, David Drakard, David Jurgens, Debajyoti Datta, Deep Ganguli, Denis Emelin, Denis Kleyko, Deniz Yuret, Derek Chen, Derek Tam, Dieuwke Hupkes, Diganta Misra, Dilyar Buzan, Dimitri Coelho Mollo, Diyi Yang, Dong-Ho Lee, Dylan Schrader, Ekaterina Shutova, Ekin Dogus Cubuk, Elad Segal, Eleanor Hagerman, Elizabeth Barnes, Elizabeth Donoway, Ellie Pavlick, Emanuele Rodolà, Emma Lam, Eric Chu, Eric Tang, Erkut Erdem, Ernie Chang, Ethan A Chi, Ethan Dyer, Ethan Jerzak, Ethan Kim, Eunice Engefu Manyasi, Evgenii Zheltonozhskii, Fanyue Xia, Fatemeh Siar, Fernando Martínez-Plumed, Francesca Happé, Francois Chollet, Frieda Rong, Gaurav Mishra, Genta Indra Winata, Gerard de Melo, Germán Kruszewski, Giambattista Parascandolo, Giorgio Mariani, Gloria Xinyue Wang, Gonzalo Jaimovitch-Lopez, Gregor Betz, Guy Gur-Ari, Hana Galijasevic, Hannah Kim, Hannah Rashkin, Hannaneh Hajishirzi, Harsh Mehta, Hayden Bogar, Henry Francis Anthony Shevlin, Hinrich Schuetze, Hiromu Yakura, Hongming Zhang, Hugh Mee Wong, Ian Ng, Isaac Noble, Jaap Jumelet, Jack Geissinger, Jackson Kernion, Jacob Hilton, Jaehoon Lee, Jaime Fernández Fisac, James B Simon, James Koppel, James Zheng, James Zou, Jan Kocon, Jana Thompson, Janelle Wingfield, Jared Kaplan, Jarema Radom, Jascha Sohl-Dickstein, Jason Phang, Jason Wei, Jason Yosinski, Jekaterina Novikova, Jelle Bosscher, Jennifer Marsh, Jeremy Kim, Jeroen Taal, Jesse Engel, Jesujoba Alabi, Jiacheng Xu, Jiaming Song, Jillian Tang, Joan Waweru, John Burden, John Miller, John U. Balis, Jonathan Batchelder, Jonathan Berant, Jörg Frohberg, Jos Rozen, Jose Hernandez-Orallo, Joseph Boudeman, Joseph Guerr, Joseph Jones, Joshua B. Tenenbaum, Joshua S. Rule, Joyce Chua, Kamil Kanclerz, Karen Livescu, Karl Krauth, Karthik Gopalakrishnan, Katerina Ignatyeva, Katja Markert, Kaustubh Dhole, Kevin Gimpel, Kevin Omondi, Kory Wallace Mathewson, Kristen Chiafullo, Ksenia Shkaruta, Kumar Shridhar, Kyle McDonell, Kyle Richardson, Laria Reynolds, Leo Gao, Li Zhang, Liam Dugan, Lianhui Qin, Lidia Contreras-Ochando, Louis-Philippe Morency, Luca Moschella, Lucas Lam, Lucy Noble, Ludwig Schmidt, Luheng He, Luis Oliveros-Colón, Luke Metz, Lütfi Kerem Senel, Maarten Bosma, Maarten Sap, Maartje Ter Hoeve, Maheen Farooqi, Manaal Faruqui, Mantas Mazeika, Marco Baturan, Marco Marelli, Marco Maru, Maria Jose Ramirez-Quintana, Marie Tolkiehn, Mario Giulianelli, Martha Lewis, Martin Potthast, Matthew L Leavitt, Matthias Hagen, Mátyás Schubert, Medina Orduna Baitemirova, Melody Arnaud, Melvin McElrath, Michael Andrew Yee, Michael Cohen, Michael Gu, Michael Ivanitskiy, Michael Starritt, Michael Strube, Michał Swędrowski, Michele Bevilacqua, Michihiro Yasunaga, Mihir Kale, Mike Cain, Mimee Xu, Mirac Suzgun, Mitch Walker, Mo Tiwari, Mohit Bansal, Moin Aminnaseri, Mor Geva, Mozhdeh Gheini, Mukund Varma T, Nanyun Peng, Nathan Andrew Chi, Nayeon Lee, Neta Gur-Ari Krakover, Nicholas Cameron, Nicholas Roberts, Nick Doiron, Nicole Martinez, Nikita Nangia, Niklas Deckers, Niklas Muennighoff, Nitish Shirish Keskar, Niveditha S. Iyer, Noah Constant, Noah Fiedel, Nuan Wen, Oliver Zhang, Omar Agha, Omar Elbaghdadi, Omer Levy, Owain Evans, Pablo Antonio Moreno Casares, Parth Doshi, Pascale Fung, Paul Pu Liang, Paul Vicol, Pegah Alipoormolabashi, Peiyuan Liao, Percy Liang, Peter W Chang, Peter Eckersley, Phu Mon Htut, Pinyu Hwang, Piotr Miłkowski, Piyush Patil, Pouya Pezeshkpour, Priti Oli, Qiaozhu Mei, Qing Lyu, Qinlang Chen, Rabin Banjade, Rachel Etta Rudolph, Raefer Gabriel, Rahel Habacker, Ramon Risco, Raphaël Millière, Rhythm Garg, Richard Barnes, Rif A. Saurous, Riku Arakawa, Robbe Raymaekers, Robert Frank, Rohan Sikand, Roman Novak, Roman Sitelew, Ronan Le Bras, Rosanne Liu, Rowan Jacobs, Rui Zhang, Russ Salakhutdinov, Ryan Andrew Chi, Seungjae Ryan Lee, Ryan Stovall, Ryan Teehan, Rylan Yang, Sahib Singh, Saif M. Mohammad, Sajant Anand, Sam Dillavou, Sam Shleifer, Sam Wiseman, Samuel Gruetter, Samuel R. Bowman, Samuel Stern Schoenholz, Sanghyun Han, Sanjeev Kwatra, Sarah A. Rous, Sarik Ghazarian, Sayan Ghosh, Sean Casey, Sebastian Bischoff, Sebastian Gehrmann, Sebastian Schuster, Sepideh Sadeghi, Shadi Hamdan, Sharon Zhou, Shashank Srivastava, Sherry Shi, Shikhar Singh, Shima Asaadi, Shixiang Shane Gu, Shubh Pachchigar, Shubham Toshniwal, Shyam Upadhyay, Shyamolima Shammie Debnath, Siamak Shakeri, Simon Thormeyer, Simone Melzi, Siva Reddy, Sneha Priscilla Makini, Soo-Hwan Lee, Spencer Torene, Sriharsha Hatwar, Stanislas Dehaene, Stefan Divic, Stefano Ermon, Stella Biderman, Stephanie Lin, Stephen Prasad, Steven Piantadosi, Stuart Shieber, Summer Misherghi, Svetlana Kiritchenko, Swaroop Mishra, Tal Linzen, Tal Schuster, Tao Li, Tao Yu, Tariq Ali, Tatsunori Hashimoto, Te-Lin Wu, Théo Desbordes, Theodore Rothschild, Thomas Phan, Tianle Wang, Tiberius Nkinyili, Timo Schick, Timofei Kornev, Titus Tunduny, Tobias Gerstenberg, Trenton Chang, Trishala Neeraj, Tushar Khot, Tyler Shultz, Uri Shaham, Vedant Misra, Vera Demberg, Victo-

ria Nyamai, Vikas Raunak, Vinay Venkatesh Ramasesh, vinay uday prabhu, Vishakh Padmakumar, Vivek Srikumar, William Fedus, William Saunders, William Zhang, Wout Vossen, Xiang Ren, Xiaoyu Tong, Xinran Zhao, Xinyi Wu, Xudong Shen, Yadollah Yaghoobzadeh, Yair Lakretz, Yangqiu Song, Yasaman Bahri, Yejin Choi, Yichi Yang, Yiding Hao, Yifu Chen, Yonatan Belinkov, Yu Hou, Yufang Hou, Yuntao Bai, Zachary Seid, Zhuoye Zhao, Zijian Wang, Zijie J. Wang, Zirui Wang, and Ziyi Wu. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. 2024. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170, Online. Association for Computational Linguistics.

Shicheng Tan, Weng Lam Tam, Yuanchun Wang, Wenwen Gong, Shu Zhao, Peng Zhang, and Jie Tang. 2023. Are intermediate layers and labels really necessary? a general language model distillation method. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9678–9696, Toronto, Canada. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, D. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, A. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. 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, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu 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*, abs/2307.09288.

Fred Tung and Greg Mori. 2019. Similarity-preserving knowledge distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1365–1374.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, Online. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020a. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NeurIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020b. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems (NeurIPS)*.

Zhecan Wang, Noel Codella, Yen-Chun Chen, Luowei Zhou, Xiyang Dai, Bin Xiao, Jianwei Yang, Haoxuan You, Kai-Wei Chang, Shih-fu Chang, et al. 2022. Multimodal adaptive distillation for leveraging unimodal encoders for vision-language tasks. *arXiv preprint arXiv:2204.10496*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

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

Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. 2023. Outlier suppression+: Accurate quantization of large language models by equivalent and effective shifting and scaling. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1648–1665, Singapore. Association for Computational Linguistics.

Xiuying Wei, Yunchen Zhang, Xiangguo Zhang, Ruihao Gong, Shanghang Zhang, Qi Zhang, Fengwei Yu, and Xianglong Liu. 2022c. Outlier suppression: Pushing the limit of low-bit transformer language models. *Advances in Neural Information Processing Systems*, 35:17402–17414.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing. *Preprint*, arXiv:1910.03771.

Haiyan Wu, Yuting Gao, Yinqi Zhang, Shaohui Lin, Yuan Xie, Xing Sun, and Ke Li. 2022a. Self-supervised models are good teaching assistants for vision transformers. In *ICML*.

Kan Wu, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan. 2022b. Tinyvit: Fast pretraining distillation for small vision transformers. In *ECCV*, pages 68–85. Springer.

Minghao Wu, Abdul Waheed, Chiyu Zhang, Muhammad Abdul-Mageed, and Alham Aji. 2024a. LaMini-LM: A diverse herd of distilled models from large-scale instructions. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 944–964, St. Julian's, Malta. Association for Computational Linguistics.

Qianhui Wu, Huiqiang Jiang, Haonan Yin, Börje F. Karlsson, and Chin-Yew Lin. 2023a. Multi-level knowledge distillation for out-of-distribution detection in text. In *The 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*.

Siyue Wu, Hongzhan Chen, Xiaojun Quan, Qifan Wang, and Rui Wang. 2023b. Ad-kd: Attribution-driven knowledge distillation for language model compression. In *Annual Meeting of the Association for Computational Linguistics*.

Taiqiang Wu, Cheng Hou, Shanshan Lao, Jiayi Li, Ngai Wong, Zhe Zhao, and Yujiu Yang. 2024b. Weight-inherited distillation for task-agnostic BERT compression. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 13–28, Mexico City, Mexico. Association for Computational Linguistics.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. 2020. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Zhendong Yang, Zhe Li, Xiaohu Jiang, Yuan Gong, Zehuan Yuan, Danpei Zhao, and Chun Yuan. 2022. Focal and global knowledge distillation for detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4643–4652.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Sergey Zagoruyko and Nikos Komodakis. 2017. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *ICLR*.

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

Tianyang Zhao, Kunwar Yashraj Singh, Srikar Appalaraju, Peng Tang, Vijay Mahadevan, R. Manmatha, and Ying Nian Wu. 2024. No head left behind – multi-head alignment distillation for transformers. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(7):7514–7524.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2023. A survey on model compression for large language models. *Preprint*, arXiv:2308.07633.

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 *The IEEE International Conference on Computer Vision (ICCV)*.

## A Appendix: Further Related Works

**Emergent Outliers in Pre-trained Language Transformers.** Although previous studies have shown that Transformers (Vaswani et al., 2017) are robust to pruning (Michel et al., 2019; Ganesh et al., 2021), Kovaleva et al. (2021); Puccetti et al. (2022) show that, in contrary, pretrained Transformers are surprisingly fragile to the removal of a very small number of particular features in the layer outputs. They observe this phenomena in BERT-family models, including BERT (Devlin et al., 2019), BART (Lewis et al., 2020), XLNet (Yang et al., 2019), ELECTRA (Clark et al., 2020); and also in GPT-2 model (Radford et al., 2019). They identify that these features are outliers of the scaling factors and biases in Layer-Norm of high-magnitude. Kovaleva et al. (2021) show that, in BERT, these outliers emerge during pretraining and remain in the same dimensional position throughout the model. Puccetti et al. (2022) further show that, in BERT, the magnitudes of these outlier dimensions correlate with the frequency of tokens in the pretraining corpus, and they also contribute to the self-attention pattern to focus on some special tokens. These two works both study outliers in Transformers at the scale around 100M parameters. Dettmers et al. (2022) scale up the study of outliers to 175B Decoder-only OPT models (Zhang et al., 2022), and observe that outlier magnitude and influence systematically emerge for all layers at and beyond 6.7B parameters. They propose to quantize these outliers separately for quantization precision. Wei et al. (2022c, 2023), alternatively, manage to suppress outliers for quantizing large Transformers.

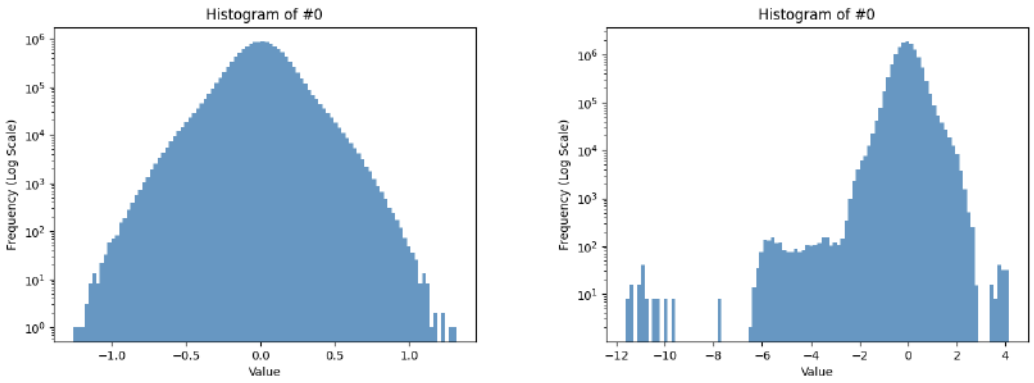
**Knowledge Distillation.** Knowledge distillation (KD) is widely applied for training more compact vision models (Zagoruyko and Komodakis, 2017; Peng et al., 2019; Tung and Mori, 2019; Yang et al., 2022; Chen et al., 2022; Wu et al., 2022b; Andonian et al., 2022; He et al., 2022; Wu et al., 2022a), language models (Sanh et al., 2019; Wang et al., 2020b; Jiao et al., 2020b; Sun et al., 2020; Liang et al., 2021; Li et al., 2022a; Ding et al., 2023b), and vision-language models (Fang et al., 2021; Wang et al., 2022; Gu et al., 2021; Ma et al., 2022; Zhao et al., 2024). With the recent emergence of large language models (Touvron et al., 2023; Anil et al., 2023; OpenAI et al., 2024; AI@Meta, 2024; Anthropic, 2024), great efforts have been made to distill some of these LLMs into smaller ones (Zhu et al., 2023; Hsieh et al., 2023; Magister et al., 2023; Wu et al., 2023a; Tan et al., 2023; Kang et al., 2023; Jha et al., 2023; Li et al., 2023; Jiang et al., 2023; Wu et al., 2024a; Gu et al., 2024; Agarwal et al., 2024), for faster inference, lower memory footprint, and lower cost.

## B Appendix: Detailed T5 Performance with Standard Error

## C Appendix: Activation Magnitude Histograms for Distilled BERT Models

| Model | Configuration | MNLI | QNLI | RTE | SST-2 | Avg (%) |
|---|---|---|---|---|---|---|
| T5-Small | full model | 0.3544 ± 0.0048 | 0.5404 ± 0.0067 | 0.5343 ± 0.0300 | 0.7099 ± 0.0154 | 53.47 |
| | non-outlier disabled: 4 dims | 0.3554 ± 0.0048 | 0.5248 ± 0.0068 | 0.5018 ± 0.0301 | 0.7959 ± 0.0137 | 54.45 |
| | outlier disabled: 4 dims | 0.3303 ± 0.0047 | 0.4843 ± 0.0068 | 0.4729 ± 0.0301 | 0.7385 ± 0.0149 | 50.65 |
| T5-Base | full model | 0.5673 ± 0.0050 | 0.5038 ± 0.0068 | 0.6137 ± 0.0293 | 0.5734 ± 0.0168 | 56.45 |
| | non-outlier disabled: 4 dims | 0.5671 ± 0.0050 | 0.5039 ± 0.0068 | 0.6209 ± 0.0292 | 0.5757 ± 0.0167 | 56.69 |
| | outlier disabled: 4 dims | 0.5142 ± 0.0050 | 0.5028 ± 0.0068 | 0.5596 ± 0.0299 | 0.6353 ± 0.0163 | 55.30 |
| T5-Large | full model | 0.6129 ± 0.0049 | 0.5061 ± 0.0068 | 0.7978 ± 0.0242 | 0.5023 ± 0.0169 | 60.48 |
| | non-outlier disabled: 4 dims | 0.6159 ± 0.0049 | 0.5059 ± 0.0068 | 0.8014 ± 0.0240 | 0.5046 ± 0.0169 | 60.70 |
| | outlier disabled: 4 dims | 0.6122 ± 0.0049 | 0.5070 ± 0.0068 | 0.7617 ± 0.0256 | 0.5011 ± 0.0169 | 59.55 |
| T5-3B | full model | 0.5060 ± 0.0051 | 0.5717 ± 0.0067 | 0.6679 ± 0.0283 | 0.8750 ± 0.0112 | 65.51 |
| | non-outlier disabled: 4 dims | 0.5046 ± 0.0050 | 0.5783 ± 0.0067 | 0.6498 ± 0.0287 | 0.8750 ± 0.0112 | 65.19 |
| | outlier disabled: 4 dims | 0.3780 ± 0.0049 | 0.5596 ± 0.0067 | 0.6606 ± 0.0285 | 0.8498 ± 0.0121 | 61.20 |
| T5-11B | full model | 0.5703 ± 0.0050 | 0.5819 ± 0.0067 | 0.6246 ± 0.0292 | 0.8762 ± 0.0112 | 66.32 |
| | non-outlier disabled: 4 dims | 0.5530 ± 0.0050 | 0.5770 ± 0.0067 | 0.6137 ± 0.0293 | 0.8761 ± 0.0112 | 65.50 |
| | outlier disabled: 4 dims | 0.4000 ± 0.0049 | 0.4523 ± 0.0067 | 0.5307 ± 0.0300 | 0.6823 ± 0.0158 | 51.63 |

Table 5: Detailed zero-shot performance on each evaluation dataset with standard error of pretrained T5 models (Raffel et al., 2020) by model size and by whether or not with our interventions of disabled outlier/non-outlier dimensions. Corresponding performance of Flan T5 models are reported in Table 1.




(a) (b)

Figure 5: Neural activation magnitude histograms for distilled BERT models, w/o EOFD and w/ EOFD, respectively. (a) The activation magnitude histogram for $BERT_6$ after pre-training distillation by TinyBERT (Jiao et al., 2020a), without EOFD; (b) The activation magnitude histogram for our EOFD-$BERT_6$ model with Emergent Outlier Focused Distillation on the MNLI dataset. The activation magnitude histogram of the model distilled with our EOFD exibihits a more long-tailed distribution.

## D Appendix: Model and Dataset Details

The Hugging Face T5 (Raffel et al., 2020), Hugging Face Flan-T5 (Chung et al., 2024), and Hugging Face BERT (Devlin et al., 2019) models are with Apache 2.0 License http://www.apache.org/licenses/; Hugging Face GPT-2 (Radford et al., 2019) models is with modified MIT License https://github.com/openai/gpt-2/blob/master/LICENSE. The TinyBERT (Jiao et al., 2020a) model is with Apache 2.0 License.

## E Appendix: Emergent Outliers in T5 Activation Magnitude Histograms

| Model | Parameters | # layers | $d_{model}$ | $d_{ff}$ | $d_{kv}$ | # heads |
|---|---|---|---|---|---|---|
| T5-Small | 60M | 6 + 6 | 512 | 2048 | 64 | 8 |
| T5-Base | 220M | 12 + 12 | 768 | 3072 | 64 | 12 |
| T5-Large | 770M | 24 + 24 | 1024 | 4096 | 64 | 16 |
| T5-3B | 3B | 24 + 24 | 1024 | 16384 | 128 | 32 |
| T5-11B | 11B | 24 + 24 | 1024 | 65536 | 128 | 128 |
| Flan-T5-Small | 80M | 8 + 8 | 512 | 1024 | 64 | 6 |
| Flan-T5-Base | 250M | 12 + 12 | 768 | 2048 | 64 | 12 |
| Flan-T5-Large | 780M | 24 + 24 | 1024 | 2816 | 64 | 16 |
| Flan-T5-XL | 3B | 24 + 24 | 2048 | 5120 | 64 | 32 |
| Flan-T5-XXL | 11B | 24 + 24 | 4096 | 10240 | 64 | 64 |
| $BERT_4$ | 14.5M | 4 | 312 | 1200 | - | 12 |
| $BERT_6$ | 66M | 6 | 768 | 3072 | - | 12 |
| BERT-Base | 109M | 12 | 768 | 3072 | - | 12 |
| GPT-2-Medium | 345M | 24 | 1024 | 4096 | - | 16 |
| GPT-2 | 117M | 12 | 768 | 3072 | - | 12 |

Table 6: Model size variants

| **Corpus** | Task | #Train | #Dev | #Test | #Label | Metrics |
|---|---|---|---|---|---|---|
| Single-Sentence Classification (GLUE) | | | | | | |
| CoLA (Warstadt et al., 2019) | Acceptability | 8.5k | 1k | 1k | 2 | Matthews corr |
| SST-2 (Socher et al., 2013) | Sentiment | 67k | 872 | 1.8k | 2 | Accuracy |
| Pairwise Text Classification (GLUE) | | | | | | |
| MNLI (Williams et al., 2018) | NLI | 393k | 20k | 20k | 3 | Accuracy |
| RTE (Bentivogli et al., 2009) | NLI | 2.5k | 276 | 3k | 2 | Accuracy |
| QQP (Chen et al., 2017) | Paraphrase | 364k | 40k | 391k | 2 | Accuracy/F1 |
| MRPC (Dolan and Brockett, 2005) | Paraphrase | 3.7k | 408 | 1.7k | 2 | Accuracy/F1 |
| QNLI (Rajpurkar et al., 2016) | QA/NLI | 108k | 5.7k | 5.7k | 2 | Accuracy |
| Text Similarity (GLUE) | | | | | | |
| STS-B (Cer et al., 2017) | Similarity | 7k | 1.5k | 1.4k | 1 | Pearson/Spearman corr |

Table 7: Summary of the eight datasets in the GLUE benchmark. This table is revised from (Liang et al., 2023a). The License is customised at this webpage https://gluebenchmark.com/faq.

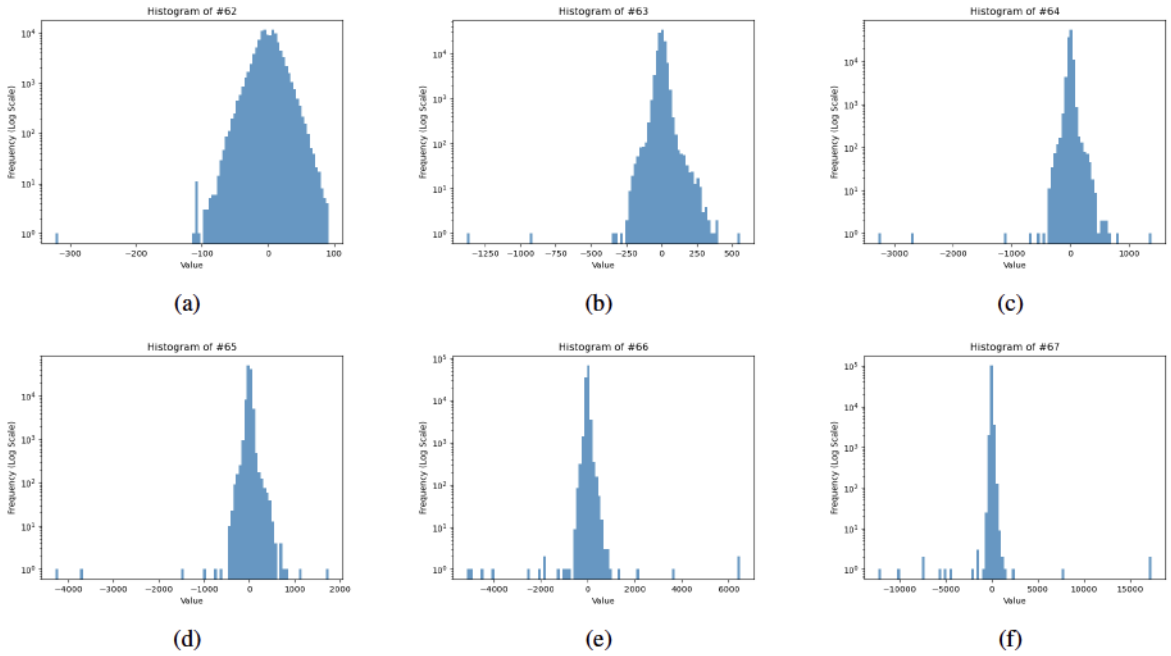



Figure 6: T5-Small Encoder hidden-state activation magnitude histograms from shallower layers to deeper layers. We visualize the 0th (embedding), 1st, 2nd, 3rd, 4th, 5th layer, respectively, in subfigure (a-f); The 6th layer is applied with an additional final LayerNorm transformation thus has a much smaller scale of magnitudes. Emergent outliers get larger in magnitudes as layer depth increases.

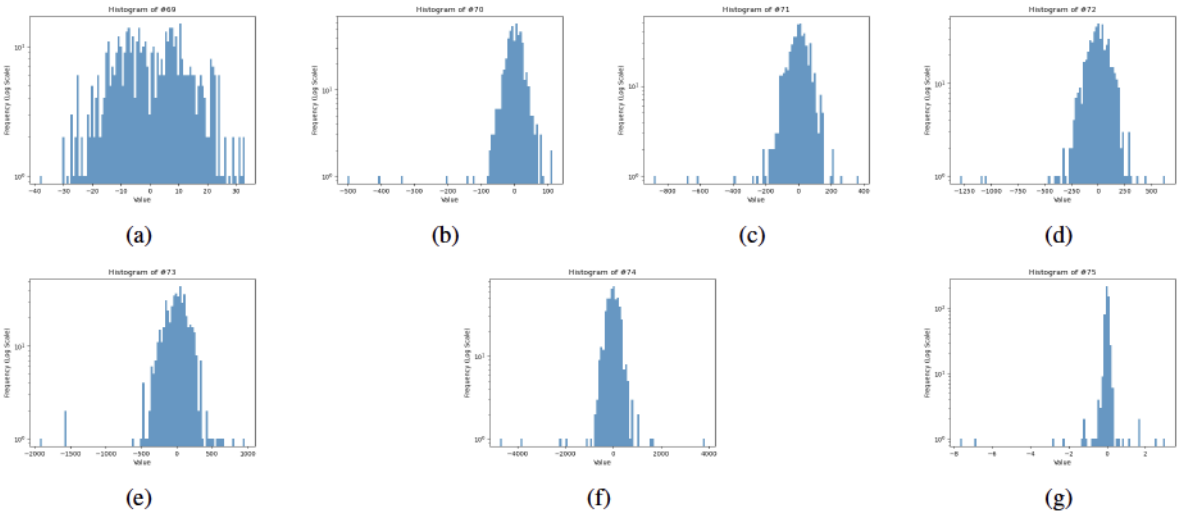



Figure 7: T5-Small Decoder hidden-state activation magnitude histograms from shallower layers to deeper layers. We visualize the 0th (embedding), 1st, 2nd, 3rd, 4th, 5th, 6th layer, respectively, in subfigure (a-g); The 6th layer is applied with an additional final LayerNorm transformation thus has a much smaller scale of magnitudes. Emergent outliers get larger in magnitudes as layer depth increases.

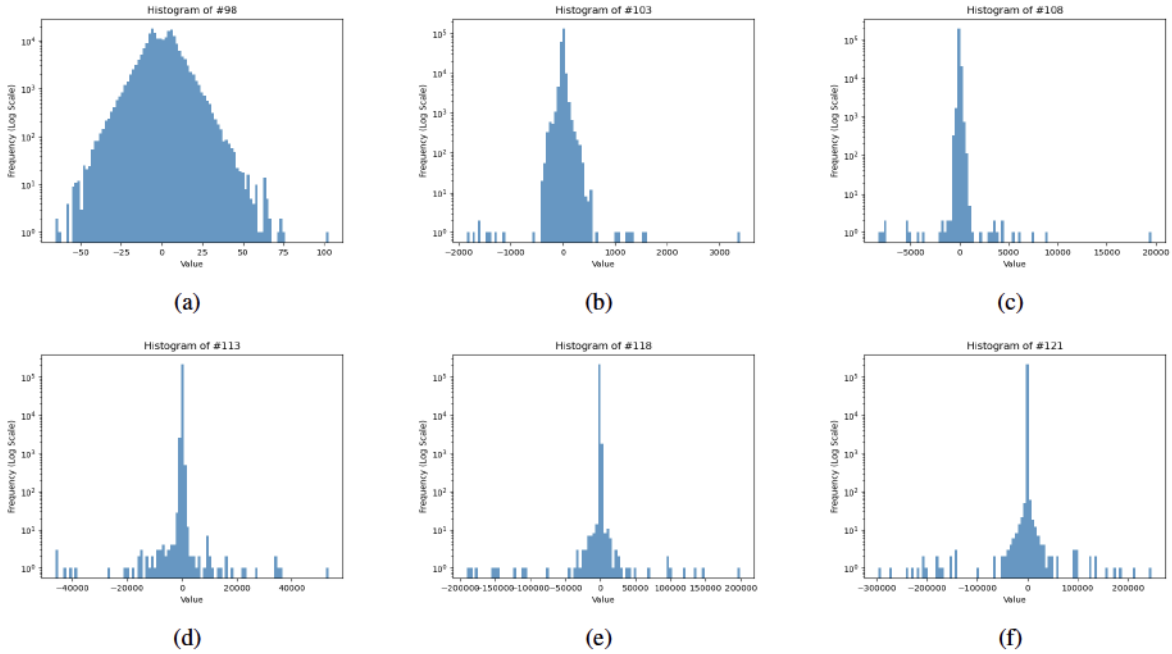



Figure 8: T5-Large Encoder hidden-state activation magnitude histograms from shallower layers to deeper layers. We visualize the 0th (embedding), 5th, 10th, 15th, 20th, 23rd layer, respectively, in subfigure (a-f); The 24th layer is applied with an additional final LayerNorm transformation thus has a much smaller scale of magnitudes. Emergent outliers get larger in magnitudes as layer depth increases.

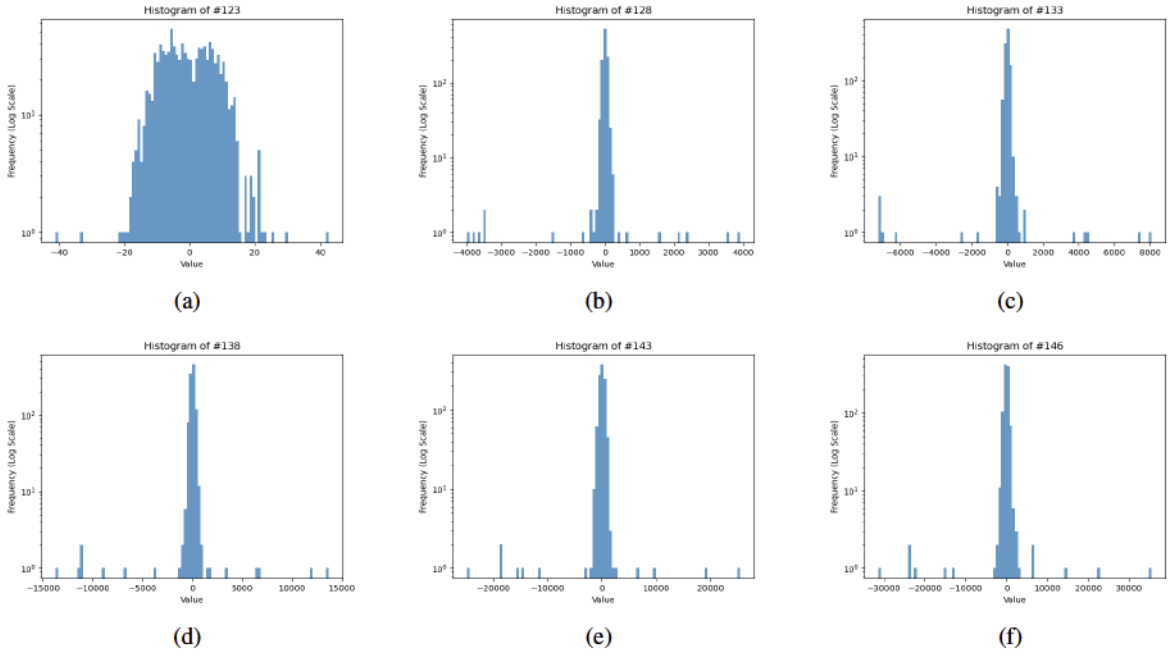



Figure 9: T5-Large Decoder hidden-state activation magnitude histograms from shallower layers to deeper layers. We visualize the 0th (embedding), 5th, 10th, 15th, 20th, 23rd layer, respectively, in subfigure (a-f); The 24th layer is applied with an additional final LayerNorm transformation thus has a much smaller scale of magnitudes. Emergent outliers get larger in magnitudes as layer depth increases.

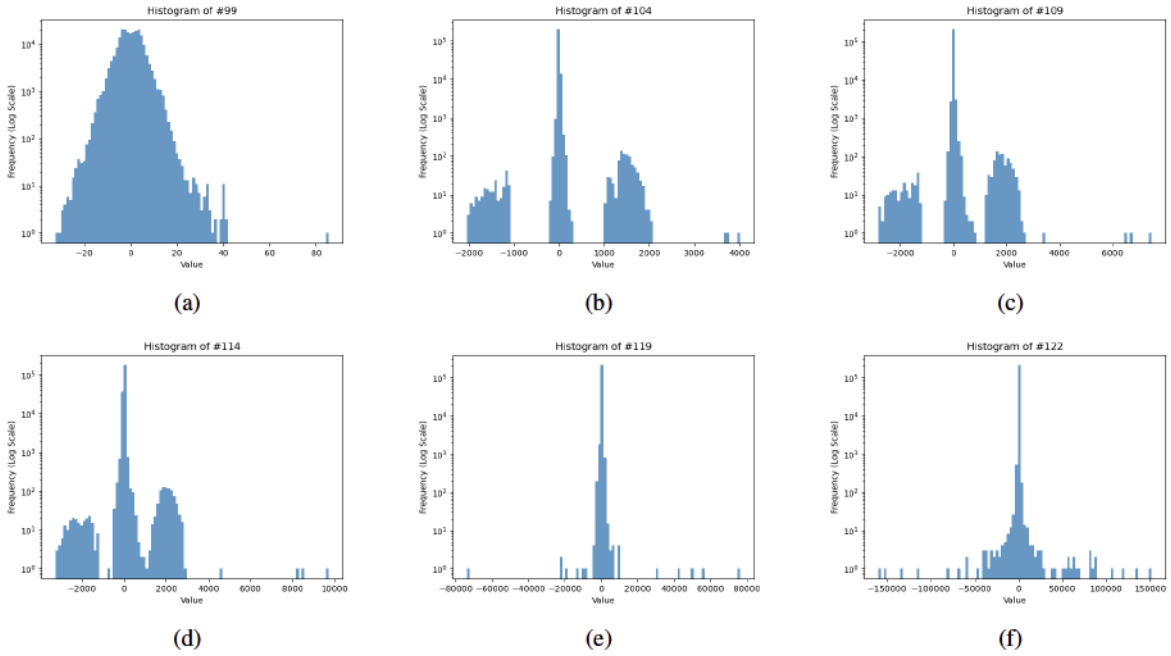



Figure 10: T5-11B Encoder hidden-state activation magnitude histograms from shallower layers to deeper layers. We visualize the 0th (embedding), 5th, 10th, 15th, 20th, 23rd layer, respectively, in subfigure (a-f); The 24th layer is applied with an additional final LayerNorm transformation thus has a much smaller scale of magnitudes. Emergent outliers get larger in magnitudes as layer depth increases.

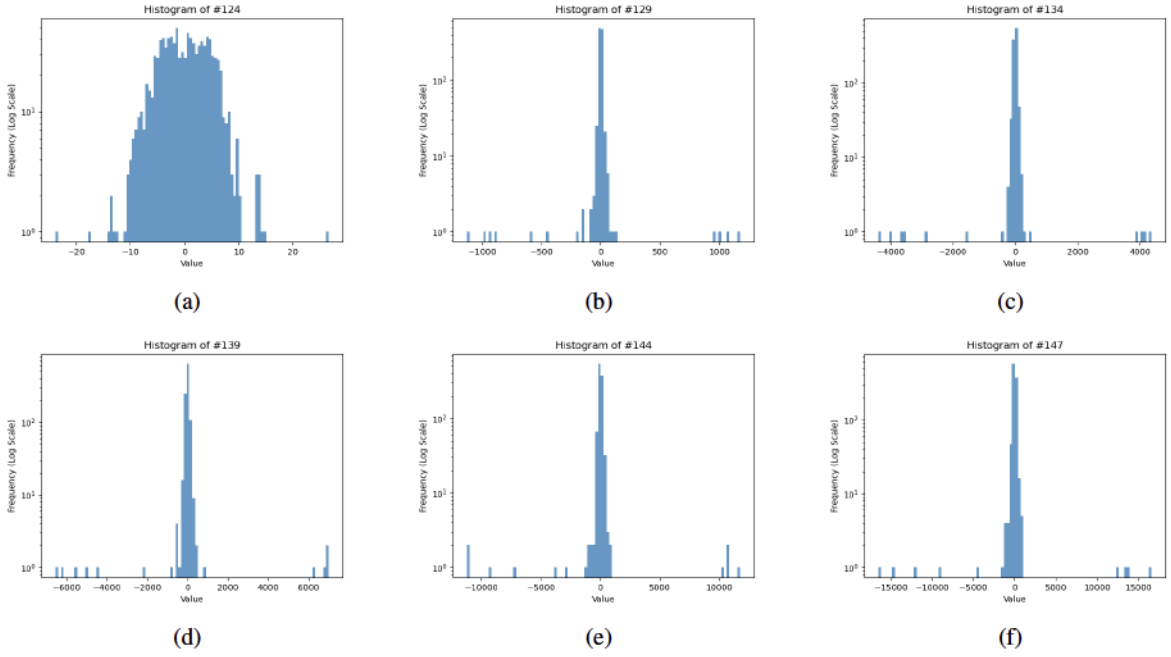



Figure 11: T5-11B Decoder hidden-state activation magnitude histograms from shallower layers to deeper layers. We visualize the 0th (embedding), 5th, 10th, 15th, 20th, 23rd layer, respectively, in subfigure (a-f); The 24th layer is applied with an additional final LayerNorm transformation thus has a much smaller scale of magnitudes. Emergent outliers get larger in magnitudes as layer depth increases. The reason why we have a larger scale of frequency in Encoders than Decoders is that in these cases, we are prompting T5 models with a Question-Answering (QA) task, where the Encoders deals with a longer sequence length than the Decoders, and hence have more tokens.

## F Appendix: Input Text For Activation Plots

question: What does increased oxygen concentrations in the patient's lungs displace? context: Hyperbaric (high-pressure) medicine uses special oxygen chambers to increase the partial pressure of O 2 around the patient and, when needed, the medical staff. Carbon monoxide poisoning, gas gangrene, and decompression sickness (the 'bends') are sometimes treated using these devices. Increased O 2 concentration in the lungs helps to displace carbon monoxide from the heme group of hemoglobin. Oxygen gas is poisonous to the anaerobic bacteria that cause gas gangrene, so increasing its partial pressure helps kill them. Decompression sickness occurs in divers who decompress too quickly after a dive, resulting in bubbles of inert gas, mostly nitrogen and helium, forming in their blood. Increasing the pressure of O 2 as soon as possible is part of the treatment.

(Target answer: carbon monoxide)

We use the above input for the activation analysis and plots. This input is one of the T5 original paper (Raffel et al., 2020) examples, without cherry-picking. The input could be found in (Raffel et al., 2020) Page 53, D.15, extracted from the SQuAD dataset (Rajpurkar et al., 2016).

## G Appendix: Training Details for Distilling BERT, T5, and GPT-2 Models

All of our code are implemented with PyTorch (Paszke et al., 2019). We conduct experiments on 8 NVIDIA A100 GPUs. We provide training details for distilling BERT, T5, and GPT-2 models in the following tables, including training both our teachers and students.

| Hyper-parameters | MNLI | QQP | QNLI | SST-2 | CoLA | RTE | MRPC | STS-B |
|---|---|---|---|---|---|---|---|---|
| Learning Rates | 7e-5 | 7e-5 | 4e-5 | 5e-5 | 3e-5 | 3e-5 | 3.5e-5 | 2e-5 |
| Batch Size | 256 | 256 | 256 | 256 | 256 | 256 | 64 | 16 |
| Training Epochs | 3 | 3 | 6 | 6 | 40 | 30 | 50 | 3 |
| Learning Rate Decay | Linear | | | | | | | |
| Learning Rate Warmup | 0 | | | | | | | |
| Max Sequence Length | 128 | | | | | | | |
| Weight Decay | 0 | | | | | | | |
| Adam $\beta_1$ | 0.9 | | | | | | | |
| Adam $\beta_2$ | 0.999 | | | | | | | |
| Adam $\epsilon$ | $1 \times 10^{-8}$ | | | | | | | |
| Gradient Clipping | 1.0 | | | | | | | |
| Initialization from MNLI | | | | | | ✓ | ✓ | |
| Random Seed | 42 | | | | | | | |

Table 8: Best hyper-parameter configurations for fine-tuning our BERT teacher models on the GLUE benchmark. We finetune by ourselves from the Hugging Face pretrained bert-base-uncased model https://huggingface.co/google-bert/bert-base-uncased with this Hugging Face PyTorch script for all datasets except the STS-B: https://github.com/huggingface/transformers/blob/main/examples/pytorch/text-classification/run_glue.py. For regression task STS-B, we use this community finetuned model instead of finetuning by ourselves: https://huggingface.co/gchhablani/bert-base-cased-finetuned-stsb. Completing each finetuning training job takes within one to two hours.

| Hyper-parameters | MNLI | QQP | QNLI | SST-2 | CoLA | RTE | MRPC | STS-B |
|---|---|---|---|---|---|---|---|---|
| Learning Rates | 8e-5 | | | | | | | |
| Batch Size | 512 | | | | | | | |
| Training Epochs | 10 | 10 | 15 | 20 | 20 | 30 | 30 | 30 |
| Learning Rate Decay | Warm-up Linear | | | | | | | |
| Learning Rate Warmup | 0.1 | | | | | | | |
| Max Sequence Length | 128 | 128 | 128 | 64 | 64 | 128 | 128 | 128 |
| Weight Decay | 1e-4 | | | | | | | |
| Adam $\beta_1$ | 0.9 | | | | | | | |
| Adam $\beta_2$ | 0.999 | | | | | | | |
| Adam $\epsilon$ | $1 \times 10^{-6}$ | | | | | | | |
| Gradient Clipping | 1.0 | | | | | | | |
| EOFD power $p$ | {0.5, 1.0} | | | | | | | |
| Initialization from MNLI | | | | | | ✓ | ✓ | ✓ |
| Random Seed | 42 | | | | | | | |

Table 9: Hyper-parameter configurations for step-1 fine-tuning distillation of our $\text{BERT}_6$ models on the GLUE benchmark. We build our code upon the open source code of TinyBERT (Jiao et al., 2020a) with minimal revisions, and we follow their procedures of dataset pre-processing, data augmentation, and the 2-step distillation pipeline for distilling all BERT models, for fair comparison. Same to them, no data augmentation is conducted on the STS-B dataset. We conduct finetuning distillation after loading their released $\text{BERT}_6$ pretrained model checkpoints for fair comparison: https://huggingface.co/huawei-noah/TinyBERT_General_6L_768D. A training job on a larger dataset like MNLI takes around a day, and a training job on a smaller dataset takes within one to two hours.

| Hyper-parameters | MNLI | QQP | QNLI | SST-2 | CoLA | RTE | MRPC | STS-B |
|---|---|---|---|---|---|---|---|---|
| Learning Rates | 3e-5 | | | | | | | |
| Batch Size | 512 | 512 | 512 | 512 | 512 | 512 | 64 | 512 |
| Training Epochs | 3 | 3 | 6 | 9 | 20 | 20 | 40 | 50 |
| Learning Rate Decay | Warm-up Linear | | | | | | | |
| Learning Rate Warmup | 0.3 | | | | | | | |
| Max Sequence Length | 128 | 128 | 128 | 64 | 64 | 128 | 128 | 128 |
| Weight Decay | 1e-4 | | | | | | | |
| Adam $\beta_1$ | 0.9 | | | | | | | |
| Adam $\beta_2$ | 0.999 | | | | | | | |
| Adam $\epsilon$ | $1 \times 10^{-6}$ | | | | | | | |
| Gradient Clipping | 1.0 | | | | | | | |
| Temperature $\tau_d$ | 1.0 | | | | | | | |
| Random Seed | 42 | | | | | | | |

Table 10: Hyper-parameter configurations for step-2 fine-tuning distillation of our $\text{BERT}_6$ models on the GLUE benchmark.

| Hyper-parameters | MNLI | QQP | QNLI | SST-2 | CoLA | RTE | MRPC | STS-B |
|---|---|---|---|---|---|---|---|---|
| Learning Rates | 7e-5 | | | | | | | |
| Batch Size | 384 | | | | | | | |
| Training Epochs | 10 | 10 | 15 | 20 | 20 | 30 | 30 | 30 |
| Learning Rate Decay | Warm-up Linear | | | | | | | |
| Learning Rate Warmup | 0.1 | | | | | | | |
| Max Sequence Length | 128 | 128 | 128 | 64 | 64 | 128 | 128 | 128 |
| Weight Decay | 1e-4 | | | | | | | |
| Adam $\beta_1$ | 0.9 | | | | | | | |
| Adam $\beta_2$ | 0.999 | | | | | | | |
| Adam $\epsilon$ | $1 \times 10^{-6}$ | | | | | | | |
| Gradient Clipping | 1.0 | | | | | | | |
| EOFD power $p$ | {0.5, 1.0} | | | | | | | |
| Initialization from MNLI | | | | | | ✓ | ✓ | ✓ |
| Random Seed | 42 | | | | | | | |

Table 11: Hyper-parameter configurations for step-1 fine-tuning distillation of our $\text{BERT}_4$ models on the GLUE benchmark. We build our code upon the open source code of TinyBERT (Jiao et al., 2020a) with minimal revisions. We conduct finetuning distillation after loading their released $\text{BERT}_4$ pretrained model checkpoint for fair comparison: https://huggingface.co/huawei-noah/TinyBERT_General_4L_312D.

| Hyper-parameters | MNLI | QQP | QNLI | SST-2 | CoLA | RTE | MRPC | STS-B |
|---|---|---|---|---|---|---|---|---|
| Learning Rates | 3e-5 | | | | | | | |
| Batch Size | 512 | 512 | 512 | 512 | 512 | 512 | 64 | 128 |
| Training Epochs | 3 | 3 | 6 | 6 | 20 | 20 | 40 | 50 |
| Learning Rate Decay | Warm-up Linear | | | | | | | |
| Learning Rate Warmup | 0.3 | | | | | | | |
| Max Sequence Length | 128 | 128 | 128 | 64 | 64 | 128 | 128 | 128 |
| Weight Decay | 1e-4 | | | | | | | |
| Adam $\beta_1$ | 0.9 | | | | | | | |
| Adam $\beta_2$ | 0.999 | | | | | | | |
| Adam $\epsilon$ | $1 \times 10^{-6}$ | | | | | | | |
| Gradient Clipping | 1.0 | | | | | | | |
| Temperature $\tau_d$ | 1.0 | | | | | | | |
| Random Seed | 42 | | | | | | | |

Table 12: Hyper-parameter configurations for step-2 fine-tuning distillation of our $\text{BERT}_4$ models on the GLUE benchmark.

| Hyper-parameters | MNLI | QQP | QNLI | CoLA |
|---|---|---|---|---|
| Learning Rates | 1e-4 | 2e-4 | 5e-5 | 1e-4 |
| Batch Size | 512 | | | |
| Training Epochs | 3 | 6 | 6 | 9 |
| Learning Rate Decay | Linear | | | |
| Learning Rate Warmup | 0 | | | |
| Max Sequence Length | 128 | | | |
| Weight Decay | 0 | | | |
| Adam $\beta_1$ | 0.9 | | | |
| Adam $\beta_2$ | 0.999 | | | |
| Adam $\epsilon$ | $1 \times 10^{-8}$ | | | |
| Gradient Clipping | 1.0 | | | |
| Random Seed | 42 | | | |

Table 13: Best hyper-parameter configurations for fine-tuning our GPT-2 Medium teacher models on the GLUE benchmark. We finetune by ourselves from the Hugging Face pretrained GPT-2-medium model https://huggingface.co/openai-community/gpt2-medium with this Hugging Face PyTorch script for all datasets listed below: https://github.com/huggingface/transformers/blob/main/examples/pytorch/text-classification/run_glue.py.

| Hyper-parameters | MNLI | QQP | QNLI | CoLA |
|---|---|---|---|---|
| Learning Rates | 7e-5 | | | |
| Batch Size | 64 | | | |
| Training Epochs | 9 | 9 | 9 | 12 |
| Hidden State Ratio $\lambda$ | 1e-4 | 1e-4 | 1e-4 | 1e-5 |
| Learning Rate Decay | Linear | | | |
| Learning Rate Warmup | 0 | | | |
| Max Sequence Length | 128 | | | |
| Weight Decay | 0 | | | |
| Adam $\beta_1$ | 0.9 | | | |
| Adam $\beta_2$ | 0.999 | | | |
| Adam $\epsilon$ | $1 \times 10^{-8}$ | | | |
| Gradient Clipping | 1.0 | | | |
| EOFD power $p$ | 0.5 | | | |
| Temperature $\tau_d$ | 1.0 | | | |
| Random Seed | 42 | | | |

Table 14: Hyper-parameter configurations for fine-tuning distillation of our GPT-2 student models on the GLUE benchmark. Our GPT-2 distillation code is revised from with this Hugging Face PyTorch script: https://github.com/huggingface/transformers/blob/main/examples/pytorch/text-classification/run_glue_no_trainer.py. We conduct finetuning distillation from the Hugging Face pretrained GPT-2 model https://huggingface.co/openai-community/gpt2. For distilling GPT and T5 models, we conduct integrated one step distillation, with the total loss of $\mathcal{L} = \mathcal{L}_{\text{PRED}} + \lambda \mathcal{L}_{\text{EOFD}}$ (or $\mathcal{L} = \mathcal{L}_{\text{PRED}} + \lambda \mathcal{L}_{\text{HID}}$, or only $\mathcal{L} = \mathcal{L}_{\text{PRED}}$, for some ablations). $\lambda$ is applied to match the scales between the losses.

| Hyper-parameters | MNLI | QQP | QNLI | SST-2 | CoLA |
|---|---|---|---|---|---|
| Learning Rates | 1.5e-4 | 4e-4 | 3e-4 | 6e-4 | 7e-4 |
| Batch Size | 256 | 512 | 512 | 512 | 512 |
| Training Epochs | 6 | 6 | 6 | 3 | 9 |
| Learning Rate Decay | Linear | | | | |
| Learning Rate Warmup | 0 | | | | |
| Max Sequence Length | 128 | | | | |
| Weight Decay | 0 | | | | |
| Adam $\beta_1$ | 0.9 | | | | |
| Adam $\beta_2$ | 0.999 | | | | |
| Adam $\epsilon$ | $1 \times 10^{-8}$ | | | | |
| Gradient Clipping | 1.0 | | | | |
| Random Seed | 42 | | | | |

Table 15: Best hyper-parameter configurations for fine-tuning our T5-base teacher models on the GLUE benchmark. We finetune by ourselves from the Hugging Face pretrained T5-base model https://huggingface.co/google-t5/t5-base with this Hugging Face PyTorch script for all datasets listed below: https://github.com/huggingface/transformers/blob/main/examples/pytorch/text-classification/run_glue.py.

| Hyper-parameters | MNLI | QQP | QNLI | SST-2 | CoLA |
|---|---|---|---|---|---|
| Learning Rates | 6e-4 | 6e-4 | 4e-4 | 3e-4 | 6e-4 |
| Batch Size | 128 | | | | |
| Training Epochs | 6 | 6 | 3 | 6 | 6 |
| Hidden State Ratio $\lambda$ | 1e-4 | | | | |
| Learning Rate Decay | Linear | | | | |
| Learning Rate Warmup | 0 | | | | |
| Max Sequence Length | 128 | | | | |
| Weight Decay | 0 | | | | |
| Adam $\beta_1$ | 0.9 | | | | |
| Adam $\beta_2$ | 0.999 | | | | |
| Adam $\epsilon$ | $1 \times 10^{-8}$ | | | | |
| Gradient Clipping | 1.0 | | | | |
| EOFD power $p$ | 0.5 | | | | |
| Temperature $\tau_d$ | 1.0 | | | | |
| Random Seed | 42 | | | | |

Table 16: Hyper-parameter configurations for fine-tuning distillation of our T5-small student models on the GLUE benchmark. Our T5 distillation code is revised from with this Hugging Face PyTorch script: https://github.com/huggingface/transformers/blob/main/examples/pytorch/text-classification/run_glue_no_trainer.py. We conduct finetuning distillation from the Hugging Face pretrained T5-small model https://huggingface.co/google-t5/t5-small. We conduct integrated one step distillation, with the total loss of $\mathcal{L} = \mathcal{L}_{\text{PRED}} + \lambda\mathcal{L}_{\text{EOFD}}$ (or $\mathcal{L} = \mathcal{L}_{\text{PRED}} + \lambda\mathcal{L}_{\text{HID}}$, or only $\mathcal{L} = \mathcal{L}_{\text{PRED}}$, for some ablations). $\lambda$ is applied to match the scales between the losses.

# H Appendix: T5 and Flan-T5 Outlier Dimensions Across Layers

## H.1 T5-Small

### H.1.1 Encoder

- Dimensions (ID) recognized as outlier dimensions for 5 times across all 6 layers: 275;
- Dimensions (ID) recognized as outlier dimensions for 4 times across all 6 layers: 42, 339;
- Dimensions (ID) recognized as outlier dimensions for 3 times across all 6 layers: 159, 324, 505;
- Dimensions (ID) recognized as outlier dimensions for 2 times across all 6 layers: 190, 260, 367;
- Dimensions (ID) recognized as outlier dimensions for 1 times across all 6 layers: 14, 23, 89, 147, 263, 264, 308, 330;

### H.1.2 Decoder

- Dimensions (ID) recognized as outlier dimensions for 5 times across all 6 layers: 206;
- Dimensions (ID) recognized as outlier dimensions for 4 times across all 6 layers: 182;
- Dimensions (ID) recognized as outlier dimensions for 2 times across all 6 layers: 245, 268, 308;
- Dimensions (ID) recognized as outlier dimensions for 1 times across all 6 layers: 1, 7, 14, 23, 31, 46, 59, 89, 98, 102, 115, 125, 137, 149, 159, 183, 190, 213, 234, 239, 294, 312, 329, 397, 410, 417, 475, 490;

## H.2 T5-11B

### H.2.1 Encoder

- Dimensions (ID) recognized as outlier dimensions for 13 times across all 24 layers: 869;
- Dimensions (ID) recognized as outlier dimensions for 5 times across all 24 layers: 680;
- Dimensions (ID) recognized as outlier dimensions for 4 times across all 24 layers: 55, 119, 165, 204, 518, 554, 607, 675, 693, 705, 753, 822, 924, 936, 1008;
- Dimensions (ID) recognized as outlier dimensions for 3 times across all 24 layers: 43, 70, 293, 295, 411, 572, 719, 925;
- Dimensions (ID) recognized as outlier dimensions for 2 times across all 24 layers: 4, 76, 155, 203, 226, 350, 512, 595, 857, 878;
- Dimensions (ID) recognized as outlier dimensions for 1 times across all 24 layers: 2, 13, 27, 141, 154, 176, 276, 396, 402, 421, 433, 435, 553, 687, 722, 810, 832, 908, 929, 968, 972;

### H.2.2 Decoder

- Dimensions (ID) recognized as outlier dimensions for 8 times across all 24 layers: 146;
- Dimensions (ID) recognized as outlier dimensions for 7 times across all 24 layers: 201, 321, 894;
- Dimensions (ID) recognized as outlier dimensions for 6 times across all 24 layers: 470, 913;
- Dimensions (ID) recognized as outlier dimensions for 5 times across all 24 layers: 109, 443, 575;
- Dimensions (ID) recognized as outlier dimensions for 4 times across all 24 layers: 53, 189, 247, 476, 632, 862, 1015;
- Dimensions (ID) recognized as outlier dimensions for 3 times across all 24 layers: 68, 84, 98, 119, 124, 210, 268, 327, 360, 432, 515, 526, 645, 650, 677, 757, 924, 1008;
- Dimensions (ID) recognized as outlier dimensions for 2 times across all 24 layers: 12, 30, 65, 72, 149, 179, 193, 248, 260, 276, 415, 423, 510, 536, 581, 584, 618, 735, 788, 805, 814, 849, 854, 863, 907, 969, 975, 994, 1014;
- Dimensions (ID) recognized as outlier dimensions for 1 times across all 24 layers: 1, 33, 49, 57, 80, 85, 97, 106, 123, 128, 137, 165, 166, 178, 188, 194, 197, 207, 230, 243, 246, 258, 262, 299, 315, 329, 351, 362, 366, 367, 379, 387, 390, 392, 395, 400, 406, 414, 424, 431, 437, 438, 440, 448, 450, 471, 474, 492, 493, 499, 513, 532, 546, 550, 568, 577, 605, 609, 616, 627, 635, 641, 643, 649, 676, 680, 682, 686, 691, 703, 708, 726, 727, 733, 734, 738, 756, 789, 793, 797, 804, 818, 823, 824, 827, 842, 853, 871, 891, 893, 899, 910, 914, 917, 934, 938, 956, 993, 1001;

## H.3 Flan-T5-Small

### H.3.1 Encoder

- Dimensions (ID) recognized as outlier dimensions for 8 times across all 8 layers: 136;
- Dimensions (ID) recognized as outlier dimensions for 7 times across all 8 layers: 511;
- Dimensions (ID) recognized as outlier dimensions for 6 times across all 8 layers: 32;
- Dimensions (ID) recognized as outlier dimensions for 5 times across all 8 layers: 163, 414;
- Dimensions (ID) recognized as outlier dimensions for 4 times across all 8 layers: 367;
- Dimensions (ID) recognized as outlier dimensions for 3 times across all 8 layers: 6, 11, 78, 412;
- Dimensions (ID) recognized as outlier dimensions for 1 times across all 8 layers: 64;

### H.3.2 Decoder

- Dimensions (ID) recognized as outlier dimensions for 7 times across all 8 layers: 247, 511;
- Dimensions (ID) recognized as outlier dimensions for 6 times across all 8 layers: 122, 396;

- Dimensions (ID) recognized as outlier dimensions for 5 times across all 8 layers: 0, 231, 242, 428, 473;
- Dimensions (ID) recognized as outlier dimensions for 4 times across all 8 layers: 67, 72;
- Dimensions (ID) recognized as outlier dimensions for 3 times across all 8 layers: 389, 456;
- Dimensions (ID) recognized as outlier dimensions for 2 times across all 8 layers: 15, 70, 97, 173, 233, 280, 305, 479;
- Dimensions (ID) recognized as outlier dimensions for 1 times across all 8 layers: 33, 45, 132, 136, 163, 175, 193, 240, 246, 268, 275, 276, 288, 329, 385, 393, 401, 415, 417;

## H.4 Flan-T5-XXL

### H.4.1 Encoder

- Dimensions (ID) recognized as outlier dimensions for 12 times across all 24 layers: 2696;
- Dimensions (ID) recognized as outlier dimensions for 11 times across all 24 layers: 1463;
- Dimensions (ID) recognized as outlier dimensions for 9 times across all 24 layers: 456;
- Dimensions (ID) recognized as outlier dimensions for 8 times across all 24 layers: 248, 2313, 2463, 2830, 2833, 3001;
- Dimensions (ID) recognized as outlier dimensions for 7 times across all 24 layers: 34, 1072, 1284, 1845, 3898;
- Dimensions (ID) recognized as outlier dimensions for 6 times across all 24 layers: 297, 1012, 1327, 1988, 2283, 2680, 2707, 3789;
- Dimensions (ID) recognized as outlier dimensions for 5 times across all 24 layers: 854, 979, 1028, 1202, 2303, 3046;
- Dimensions (ID) recognized as outlier dimensions for 4 times across all 24 layers: 295, 379, 586, 792, 900, 1301, 1583, 2795, 3002;
- Dimensions (ID) recognized as outlier dimensions for 3 times across all 24 layers: 181, 739, 1154, 1478, 1601, 1891, 1941, 2020, 2023, 2082, 2218, 2508, 2538, 2673, 2775, 3266, 3651, 3660, 3766, 3909, 3982;
- Dimensions (ID) recognized as outlier dimensions for 2 times across all 24 layers: 19, 35, 124, 251, 540, 749, 998, 1208, 1291, 1599, 1776, 2175, 2244, 2826, 3107, 3152, 3483, 3624, 3633, 3743, 3907;
- Dimensions (ID) recognized as outlier dimensions for 1 times across all 24 layers: 105, 168, 171, 204, 206, 244, 285, 307, 418, 431, 465, 527, 579, 595, 645, 826, 892, 912, 924, 994, 997, 1024, 1093, 1099, 1107, 1116, 1158, 1258, 1375, 1376, 1392, 1430, 1544, 1563, 1589, 1597, 1603, 1640, 1684, 1702, 1741, 1747, 1788, 1850, 1895, 1901, 1964, 2001, 2018, 2076, 2142, 2144, 2149, 2162, 2167, 2200, 2212, 2321, 2405, 2417, 2452, 2491, 2612, 2698, 2730, 2816, 2845, 2861, 2874, 2891, 2958, 2993, 2999, 3025, 3037, 3153, 3163, 3222, 3240, 3285, 3296, 3327, 3353, 3372, 3509, 3594, 3702, 3774, 3778, 3782, 3797, 3801, 3857, 3873, 3940, 3951, 3961, 3968, 3980, 3991;

### H.4.2 Decoder

- Dimensions (ID) recognized as outlier dimensions for 23 times across all 24 layers: 550;
- Dimensions (ID) recognized as outlier dimensions for 21 times across all 24 layers: 3280;
- Dimensions (ID) recognized as outlier dimensions for 20 times across all 24 layers: 2297;
- Dimensions (ID) recognized as outlier dimensions for 15 times across all 24 layers: 112, 339;
- Dimensions (ID) recognized as outlier dimensions for 14 times across all 24 layers: 3327;
- Dimensions (ID) recognized as outlier dimensions for 13 times across all 24 layers: 3874;
- Dimensions (ID) recognized as outlier dimensions for 12 times across all 24 layers: 303, 433, 2576, 3579, 3835;
- Dimensions (ID) recognized as outlier dimensions for 10 times across all 24 layers: 1426, 2257, 2316, 3604;
- Dimensions (ID) recognized as outlier dimensions for 9 times across all 24 layers: 1093, 1416, 2627;
- Dimensions (ID) recognized as outlier dimensions for 8 times across all 24 layers: 784, 961, 1154, 1310, 1421, 1799, 2008, 2339, 2724;
- Dimensions (ID) recognized as outlier dimensions for 7 times across all 24 layers: 45, 98, 109, 765, 927, 1409, 1723, 2685, 3462;
- Dimensions (ID) recognized as outlier dimensions for 6 times across all 24 layers: 703, 747, 909, 984, 1279, 1713, 3054, 3782, 3920;
- Dimensions (ID) recognized as outlier dimensions for 5 times across all 24 layers: 324, 600, 726, 877, 974, 1197, 1456, 1503, 1850, 2005, 2485, 2534, 2624, 2830, 2874, 3084, 3232, 3410, 3466, 3646, 3866, 3964;
- Dimensions (ID) recognized as outlier dimensions for 4 times across all 24 layers: 19, 410, 709, 962, 1088, 1658, 1760, 1900, 2299, 2493, 2514, 2693, 2938, 3158, 3169, 3588, 3627, 3903, 3917, 4027, 4072;
- Dimensions (ID) recognized as outlier dimensions for 3 times across all 24 layers: 257, 277, 489, 553, 628, 668, 831, 869, 876, 891, 1048, 1067, 1079, 1175, 1248, 1320, 1396, 1413, 1438, 1529, 1566, 1665, 1813, 1901, 1911, 1915, 1945, 2035, 2143, 2158, 2247, 2337, 2588, 2633, 2636, 2669, 2800, 2904, 2905, 3015, 3020, 3050, 3061, 3069, 3216, 3464, 3488, 3491, 3562, 3584, 3656, 3691, 3824, 3862, 3902, 3962, 3992;
- Dimensions (ID) recognized as outlier dimensions for 2 times across all 24 layers: 10, 18, 139, 206, 256, 265, 300, 328, 359, 367, 401, 418, 445, 448, 575, 608, 660, 661, 795, 819, 851, 856, 898, 943, 950, 968, 1098, 1107, 1116, 1198, 1232, 1259, 1302, 1441, 1442, 1472, 1505, 1519, 1534, 1597, 1624, 1642, 1686, 1712, 1740, 1816, 1825, 1849, 1853, 1887, 2054, 2079, 2080, 2081, 2122, 2142, 2146, 2205, 2232, 2236, 2255, 2336, 2358, 2452, 2464, 2500, 2556, 2563, 2665, 2705, 2711, 2721, 2804, 2815, 2833, 2859, 2964, 3138, 3192, 3207, 3217, 3226, 3228, 3285, 3294, 3338, 3342, 3397, 3398, 3418, 3426, 3553, 3624, 3818, 3836, 3855, 3886, 3913, 4000, 4036, 4051, 4063, 4080;

- Dimensions (ID) recognized as outlier dimensions for 1 time across all 24 layers: 0, 4, 14, 26, 28, 48, 50, 52, 53, 55, 57, 81, 91, 108, 113, 114, 121, 131, 146, 156, 195, 199, 219, 240, 248, 250, 295, 302, 340, 363, 372, 376, 406, 411, 434, 437, 441, 454, 455, 467, 471, 475, 476, 487, 498, 526, 527, 528, 541, 566, 584, 596, 658, 669, 686, 697, 706, 719, 728, 743, 766, 772, 798, 821, 837, 844, 847, 884, 892, 914, 937, 940, 942, 948, 969, 973, 979, 983, 993, 998, 1009, 1024, 1038, 1039, 1043, 1044, 1069, 1081, 1082, 1111, 1131, 1137, 1169, 1177, 1178, 1185, 1191, 1285, 1298, 1303, 1314, 1315, 1336, 1344, 1350, 1364, 1372, 1381, 1399, 1412, 1429, 1435, 1447, 1466, 1478, 1481, 1506, 1511, 1526, 1527, 1540, 1545, 1556, 1580, 1585, 1586, 1608, 1614, 1631, 1633, 1646, 1667, 1683, 1699, 1700, 1706, 1721, 1734, 1745, 1768, 1785, 1787, 1793, 1794, 1817, 1824, 1826, 1837, 1843, 1846, 1847, 1865, 1867, 1874, 1885, 1917, 1929, 1952, 2001, 2007, 2015, 2017, 2039, 2040, 2060, 2071, 2083, 2090, 2100, 2109, 2112, 2115, 2124, 2170, 2187, 2190, 2211, 2214, 2218, 2243, 2245, 2259, 2262, 2278, 2287, 2300, 2315, 2320, 2325, 2354, 2357, 2372, 2374, 2378, 2390, 2405, 2470, 2496, 2501, 2530, 2569, 2586, 2587, 2590, 2607, 2626, 2652, 2660, 2670, 2677, 2684, 2689, 2690, 2694, 2696, 2698, 2708, 2712, 2714, 2722, 2743, 2748, 2751, 2777, 2791, 2792, 2803, 2810, 2818, 2837, 2856, 2863, 2876, 2887, 2898, 2907, 2917, 2918, 2947, 2989, 3016, 3025, 3036, 3075, 3078, 3085, 3086, 3094, 3106, 3107, 3121, 3154, 3172, 3173, 3179, 3182, 3202, 3211, 3227, 3240, 3246, 3252, 3313, 3318, 3322, 3353, 3367, 3412, 3435, 3436, 3452, 3453, 3467, 3481, 3495, 3535, 3547, 3559, 3567, 3569, 3577, 3586, 3609, 3637, 3638, 3670, 3692, 3704, 3712, 3717, 3718, 3719, 3721, 3739, 3744, 3746, 3747, 3753, 3775, 3793, 3794, 3806, 3810, 3821, 3825, 3841, 3879, 3880, 3882, 3892, 3910, 3929, 3931, 3976, 3980, 3982, 4028, 4056, 4058, 4061, 4077;

# I Appendix: Code for the Emergent Outlier Focused Distillation Loss

```python
def outlier_focused_distillation_loss(student, teacher, power=0.5):
    # Compute EOFD loss for a given layer l.
    # Input: batched teacher and student intermediate representation
    #     (H^t_l and H^s_l) tensors of shape (B, N, d_t or d_s)
    # Input: EOFD power p

    weights = calculate_weight(teacher, power=power)

    squared_diff = (student - teacher) ** 2
    weighted_squared_diff = squared_diff * weights
    weighted_mse_loss = weighted_squared_diff.mean()
    return weighted_mse_loss

def calculate_weight(teacher, power):
    # teacher tensor of shape (B, N, d_t)
    (batch_size, sequence_length, hidden_size) = teacher.shape

    std_hidden = teacher.std(dim=(0, 1))
        # of shape (d_t)
    mean_std = std_hidden.mean()

    std_scaled = std_hidden / mean_std
        # of shape (d_t)

    # Scaling, if power=0, should be vanilla MSE loss
    std_scaled = std_scaled ** power
        # of shape (d_t)

    # Broadcasting the new weights to the original shape
    std_scaled_unsqueezed = std_scaled.unsqueeze(0).unsqueeze(0)
        # of shape (1, 1, d_t)
    weights = std_scaled_unsqueezed.expand(batch_size, \
                sequence_length, hidden_size)
        # of shape (B, N, d_t)

    return weights
```

Figure 12: PyTorch Code for the proposed Emergent Outlier Focused Distillation (EOFD) Loss.