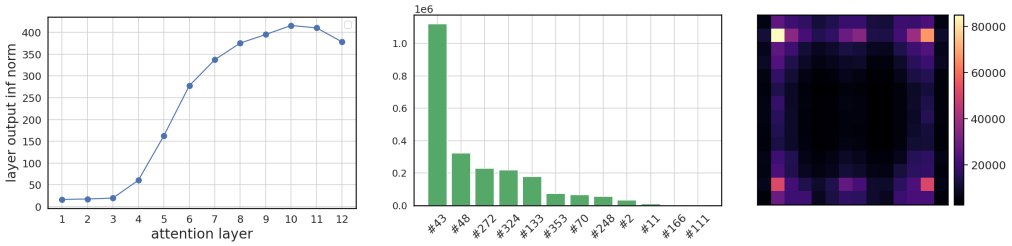
Supplementary materials for Quantizable Transformers: Removing Outliers by Helping Attention Heads Do Nothing

Anonymous Author(s)
Affiliation
Address
email

A Additional graphs from outlier analysis

In this section, we present additional graphs from our outlier investigation in Section 3 for BERT and vision transformer.



(a) (b) (c)

Figure 1: A summary of several outlier statistics recorded from ImageNet validation set on ViT. (a) Average infinity norm of the output of each attention layer. (b) A histogram of outlier counts in attention layer #10 vs. hidden dimensions. We use zero-based indexing for dimensions. (c) A heatmap of outlier counts in attention layer #10 vs. patch positions.

BERT Recall from Figure 1 that all the outliers are only present in hidden dimensions #123, #180, #225, #308, #381, #526, #720 (with the majority of them in #180, #720). These hidden dimensions correspond to attention heads #2, #3, #4, #5, #6, #9, and #12. In Figures 9 and 10 we show more examples of the discovered self-attention patterns for attention heads #3 and #12 ($\leftrightarrow$ hidden dim #180 and #720, respectively). We also show self-attention patterns in attention heads and layers which are not associated with the outliers in Figures 11 and 12, respectively.

ViT Figure 8 further shows that there are a lot of similarities in the outlier behavior in the vision transformer, compared to BERT. The strongest magnitude outliers generally happen in the later layers, peaking at layers #10 and #11. The majority of outliers ($> 99\%$) are only ever happening in only 10 hidden dimensions, notably in dimensions #48 and #43, which corresponds to the attention head #1. Finally, averaged across the entire ImageNet validation set, the outliers seem to be concentrated at the boundaries of the image, which suggest a strong correlation with the background (and a negative correlation with the object, which is usually in the center of the image in the ImageNet dataset).

In Figures 13 and 14, we show more examples of outlier and self-attention patterns in the attention head #1 ($\leftrightarrow$ hidden dimensions #48, #43) for a random subset of images from the ImageNet validation set (in layers #10 and #11, respecively).

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

B Detailed results

In this section, we provide extended results for each model, including the used hyperparameters and other design choices. We also present some additional ablation studies.

B.1 Gating architectures

Configuration	$\boldsymbol{G}$	Memory overhead (per attention layer) # extra parameters	# extra tokens
Linear	$n_{\text{heads}} \times \text{Linear}(d_{\text{head}} \to 1)$	$n_{\text{heads}}(d_{\text{head}} + 1)$	~ 1
MLP	$n_{\text{heads}} \times \text{MLP}(d_{\text{head}} \to n_{\text{hid}} \to 1)$	$n_{\text{heads}}(n_{\text{hid}}(d_{\text{head}} + 2) + 1)$	$\sim n_{\text{hid}}$
All-heads-linear	$\text{Linear}(d_{\text{model}} \to n_{\text{heads}})$	$n_{\text{heads}}(d_{\text{model}} + 1)$	$\sim n_{\text{heads}}$

Table 1: An overview of the gating function parameterizations and their memory overhead explored in this paper.

We investigate the choice of several gating functions, summarized in Table 3. The configuration "MLP" parameterizes each $\boldsymbol{G}_i$ with a feed-forward net with one hidden layer of size n_{hid} and a ReLU non-linearity [44]. We also explore what happens if we allow the mixing of the representation from different attention heads in the "All-heads-linear" setting, where we use a single linear layer to produce the gating probabilities for all attention heads at once. All three options are tested below. Unless explicitly stated otherwise, we initialize the bias of the gating function to zero (i.e., $b_{\text{init}} = 0$, $\pi_{\text{init}} = 0.5$).

B.2 BERT

Method	FP16 ppl.↓	Max inf norm	Avg. Kurtosis	W8A8 ppl.↓
Vanilla	$4.49^{\pm 0.01}$	$735.0^{\pm 54.9}$	$3076^{\pm 262}$	$1294^{\pm 1046}$
CS ($\gamma = -0.005$)	$4.44^{\pm 0.02}$	$406.6^{\pm 35.2}$	$1963^{\pm 753}$	$75.27^{\pm 39.57}$
CS ($\gamma = -0.01$)	$4.35^{\pm 0.01}$	$198.3^{\pm 78.7}$	$1581^{\pm 839}$	$7.06^{\pm 2.37}$
CS ($\gamma = -0.015$)	$4.37^{\pm 0.01}$	$38.9^{\pm 7.9}$	$165^{\pm 34}$	$4.54^{\pm 0.01}$
CS ($\gamma = -0.02$)	$4.39^{\pm 0.02}$	$31.7^{\pm 6.3}$	$90^{\pm 20}$	$4.56^{\pm 0.02}$
CS ($\gamma = -0.025$)	$\mathbf{4.39^{\pm 0.00}}$	$\mathbf{21.5^{\pm 1.5}}$	$\mathbf{80^{\pm 6}}$	$\mathbf{4.52^{\pm 0.01}}$
CS ($\gamma = -0.03$)	$4.41^{\pm 0.01}$	$20.4^{\pm 0.2}$	$79^{\pm 6}$	$4.55^{\pm 0.01}$
CS ($\gamma = -0.04$)	$4.51^{\pm 0.05}$	$19.8^{\pm 9.0}$	$85^{\pm 7}$	$4.65^{\pm 0.06}$
GA, Linear ($\pi_{\text{init}} = 0.25$)	$4.49^{\pm 0.00}$	$139.8^{\pm 62.3}$	$739^{\pm 412}$	$5.05^{\pm 0.27}$
GA, Linear ($\pi_{\text{init}} = 0.5$)	$4.48^{\pm 0.00}$	$177.3^{\pm 33.2}$	$652^{\pm 81}$	$5.13^{\pm 0.15}$
GA, Linear ($\pi_{\text{init}} = 0.75$)	$4.49^{\pm 0.00}$	$71.4^{\pm 49.9}$	$262^{\pm 147}$	$4.88^{\pm 0.22}$
GA, Linear ($\pi_{\text{init}} = 0.9$)	$4.49^{\pm 0.00}$	$171.5^{\pm 8.8}$	$559^{\pm 141}$	$5.15^{\pm 0.03}$
GA, MLP ($n_{\text{hid}} = 4$)	$\mathbf{4.45^{\pm 0.03}}$	$\mathbf{39.2^{\pm 26.0}}$	$\mathbf{201^{\pm 181}}$	$\mathbf{4.65^{\pm 0.04}}$
GA, MLP ($n_{\text{hid}} = 64$)	$4.49^{\pm 0.01}$	$117.0^{\pm 48.3}$	$507^{\pm 167}$	$4.77^{\pm 0.01}$
GA, All-heads-linear	$4.49^{\pm 0.01}$	$58.3^{\pm 41.2}$	$334^{\pm 321}$	$4.67^{\pm 0.03}$

Table 2: Main results for our proposed Clipped softmax (CS) and Gated attention (GA) applied to BERT-base. We report the masked language modeling perplexity (ppl for short) on the English Wikipedia validation set (floating-point baseline and W8A8 quantized model). We also report the maximum $\|\mathbf{x}\|_\infty$ averaged across the validation set, and kurtosis of $\mathbf{x}$ averaged across all layers, where $\mathbf{x}$ is the output of an attention layer.

Detailed results for BERT-base are summarized in Table 4. As we can see, across most of the settings, both of our methods significantly dampen the outliers' magnitude, reduce the kurtosis, drastically improve the quantized performance, while maintaining and sometimes improving the FP16 perplexity.

B.3 OPT

Detailed results for OPT-125m are summarized in Table 5.

Method	LN γ wd	FP16 ppl.↓	Max inf norm	Avg. Kurtosis	W8A8 ppl.↓
Vanilla	×	$15.84^{\pm 0.05}$	$339.6^{\pm 47.2}$	$1777^{\pm 444.}$	$21.18^{\pm 1.89}$
GA, Linear ($\pi_{\text{init}} = 0.1$)	×	$15.61^{\pm 0.05}$	$35.6^{\pm 4.5}$	$42.4^{\pm 22.9}$	$16.41^{\pm 0.18}$
GA, Linear ($\pi_{\text{init}} = 0.25$)	×	$\mathbf{15.50^{\pm 0.04}}$	$\mathbf{35.8^{\pm 0.5}}$	$\mathbf{59.0^{\pm 48.3}}$	$\mathbf{16.25^{\pm 0.08}}$
GA, Linear ($\pi_{\text{init}} = 0.5$)	×	$15.54^{\pm 0.01}$	$46.5^{\pm 5.0}$	$40.6^{\pm 8.9}$	$16.30^{\pm 0.01}$
GA, All-heads-linear	×	$15.43^{\pm 0.01}$	$32.8^{\pm 1.7}$	$24.2^{\pm 3}$	$16.30^{\pm 0.12}$
Vanilla	✓	$15.96^{\pm 0.03}$	$87.7^{\pm 31.9}$	$2080^{\pm 1460}$	$39.46^{\pm 16.59}$
CS ($\gamma = -1/512$)	✓	$15.99^{\pm 0.02}$	$106.4^{\pm 7.0}$	$5764^{\pm 2150}$	$185.23^{\pm 220.00}$
CS ($\gamma = -2/512$)	✓	$15.90^{\pm 0.02}$	$102.0^{\pm 27.0}$	$11290^{\pm 4372}$	$60.90^{\pm 52.70}$
CS ($\gamma = -4/512$)	✓	$15.86^{\pm 0.01}$	$83.1^{\pm 20.6}$	$17174^{\pm 7791}$	$84.64^{\pm 10.55}$
CS ($\gamma = -8/512$)	✓	$16.13^{\pm 0.09}$	$61.5^{\pm 9.9}$	$19204^{\pm 4284}$	$42.62^{\pm 3.64}$
CS ($\gamma = -12/512$)	✓	$16.29^{\pm 0.07}$	$63.2^{\pm 8.8}$	$19727^{\pm 7479}$	$37.22^{\pm 2.39}$
GA, Linear ($\pi_{\text{init}} = 0.1$)	✓	$15.69^{\pm 0.05}$	$7.3^{\pm 0.4}$	$25.4^{\pm 10}$	$16.23^{\pm 0.08}$
GA, Linear ($\pi_{\text{init}} = 0.25$)	✓	$\mathbf{15.55^{\pm 0.05}}$	$\mathbf{8.7^{\pm 0.6}}$	$\mathbf{18.9^{\pm 1}}$	$\mathbf{16.02^{\pm 0.07}}$
GA, Linear ($\pi_{\text{init}} = 0.5$)	✓	$15.63^{\pm 0.00}$	$10.8^{\pm 0.7}$	$42.0^{\pm 19}$	$16.20^{\pm 0.01}$
GA, All-heads-linear	✓	$15.53^{\pm 0.01}$	$7.9^{\pm 0.3}$	$13.8^{\pm 1}$	$16.09^{\pm 0.08}$

Table 3: Main results for our proposed Clipped softmax (CS) and Gated attention (GA) applied to OPT-125m. We report the causal language modeling perplexity (ppl for short) on the English Wikipedia validation set (floating-point baseline and W8A8 quantized model). We also report the maximum $\|\mathbf{x}\|_\infty$ averaged across the validation set, and kurtosis of $\mathbf{x}$ averaged across all layers, where $\mathbf{x}$ is the output of an attention layer.

In our early experiments on a smaller OPT model, we found that applying the weight decay on LayerNorm weights $\boldsymbol{\gamma}$ (which isn't the case, by default) has a strong effect on reducing the outliers' magnitude while yielding the comparable FP16 performance. Therefore, we present the results of applying our gated attention approach in both cases, with and without applying weight decay on LN $\boldsymbol{\gamma}$. As we can see in Table 5, in both cases gated attention (further) dampens the outliers' magnitude to a great extent, reduces the kurtosis, and yields models with significantly higher quantized performance, which is close to the original FP16 performance.

B.4 ViT

Detailed results for ViT-S/16 are summarized in Table 6.

After our preliminary experiments on ViT, we noticed that distinct outliers already originate after the patch embeddings. Therefore, we experimented with adding the LayerNorm after the patch embeddings (which was absent in the model definition, by default). As we can see in Table 5, together with this change, both of our proposed methods greatly dampens the outliers' magnitude, reduces the kurtosis, and yields models with significantly higher quantized performance, which is within 1% of the original FP32 accuracy.

B.5 The impact of clipped softmax hyperparameters (γ and ζ) on ViT

We investigate the effect of different values of the clipped softmax stretch parameters applied to the vision transformer and present the results in Table 7. To speed up training, for this experiment we trained ViT for 150 epochs instead of the usual 300 epochs. For this experiment, we did not apply LayerNorm after the patch embeddings.

We found similar observations compared to BERT. Specifically, most of the improvement happens when we use $\gamma < 0$ (clipping at zero) whereas using $\zeta > 1$ (clipping at one) yields similar results to the vanilla softmax and combining both $\gamma < 0$ and $\zeta > 1$ yields similar results compared to just clipping at zero.

Method	Patch. Embd. LN	FP32 acc.	Max inf norm	Avg. Kurtosis	W8A8 acc.
Vanilla	×	$80.75^{\pm 0.10}$	$358.5^{\pm 81.2}$	$1018.29^{\pm 471.46}$	$69.24^{\pm 6.93}$
CS ($\gamma = -0.003$)	×	$80.24^{\pm 0.05}$	$69.3^{\pm 20.7}$	$25.56^{\pm 8.55}$	$78.71^{\pm 0.33}$
CS ($\gamma = -0.004$)	×	$80.38^{\pm 0.01}$	$74.9^{\pm 10.6}$	$30.55^{\pm 4.88}$	$78.66^{\pm 0.49}$
GA, Linear ($\pi_{\text{init}} = 0.25$)	×	$80.62^{\pm 0.01}$	$86.0^{\pm 8.0}$	$23.44^{\pm 2.69}$	$79.16^{\pm 0.05}$
GA, Linear ($\pi_{\text{init}} = 0.5$)	×	$80.32^{\pm 0.02}$	$88.4^{\pm 17.9}$	$27.89^{\pm 14.02}$	$78.90^{\pm 0.25}$
GA, MLP ($n_{\text{hid}} = 4$)	×	$80.62^{\pm 0.05}$	$118.2^{\pm 40.5}$	$47.84^{\pm 29.79}$	$78.79^{\pm 0.29}$
Vanilla	✓	$80.98^{\pm 0.08}$	$81.1^{\pm 2.5}$	$24.53^{\pm 1.79}$	$79.62^{\pm 0.06}$
CS ($\gamma = -0.0001$)	✓	$\mathbf{80.89^{\pm 0.13}}$	$\mathbf{73.7^{\pm 14.9}}$	$\mathbf{22.92^{\pm 1.57}}$	$\mathbf{79.77^{\pm 0.25}}$
CS ($\gamma = -0.0003$)	✓	$80.92^{\pm 0.07}$	$78.9^{\pm 5.5}$	$23.83^{\pm 0.49}$	$79.63^{\pm 0.05}$
CS ($\gamma = -0.0005$)	✓	$80.95^{\pm 0.08}$	$72.9^{\pm 11.8}$	$24.46^{\pm 0.70}$	$79.73^{\pm 0.08}$
CS ($\gamma = -0.001$)	✓	$80.95^{\pm 0.16}$	$80.8^{\pm 2.1}$	$24.07^{\pm 0.65}$	$79.69^{\pm 0.03}$
CS ($\gamma = -0.002$)	✓	$80.80^{\pm 0.07}$	$78.0^{\pm 0.5}$	$25.77^{\pm 0.68}$	$79.32^{\pm 0.07}$
CS ($\gamma = -0.003$)	✓	$80.79^{\pm 0.02}$	$75.6^{\pm 7.9}$	$28.09^{\pm 4.05}$	$79.00^{\pm 0.10}$
GA, Linear ($\pi_{\text{init}} = 0.5$)	✓	$\mathbf{81.01^{\pm 0.06}}$	$\mathbf{79.8^{\pm 0.5}}$	$\mathbf{19.88^{\pm 0.28}}$	$\mathbf{79.82^{\pm 0.11}}$
GA, Linear ($\pi_{\text{init}} = 0.75$)	✓	$81.01^{\pm 0.05}$	$77.8^{\pm 0.3}$	$21.80^{\pm 1.92}$	$79.80^{\pm 0.08}$
GA, Linear ($\pi_{\text{init}} = 0.9$)	✓	$80.92^{\pm 0.11}$	$70.6^{\pm 8.0}$	$23.19^{\pm 3.74}$	$79.64^{\pm 0.09}$

Table 4: Main results for our proposed Clipped softmax (CS) and Gated attention (GA) applied to ViT-S/16. We report the top-1 accuracy on ImageNet-1K validation set for floating-point baseline and W8A8 quantized model. We also report the maximum $\|\mathbf{x}\|_\infty$ averaged across the validation set, and kurtosis of $\mathbf{x}$ averaged across all layers, where $\mathbf{x}$ is the output of an attention layer.

γ	ζ	FP32 acc.	Max inf norm	W8A8 acc.
0 (= Vanilla)	1	$78.80^{\pm 0.42}$	$426^{\pm 69}$	$71.27^{\pm 0.88}$
0	1.001	$78.78^{\pm 0.29}$	$411^{\pm 88}$	$71.24^{\pm 0.59}$
0	1.002	$78.90^{\pm 0.17}$	$420^{\pm 47}$	$70.74^{\pm 0.34}$
0	1.004	$78.80^{\pm 0.45}$	$377^{\pm 67}$	$72.31^{\pm 0.06}$
0	1.01	$78.81^{\pm 0.30}$	$419^{\pm 77}$	$71.35^{\pm 0.26}$
−0.00001	1	$78.81^{\pm 0.21}$	$432^{\pm 76}$	$69.02^{\pm 0.19}$
−0.0001	1	$78.81^{\pm 0.36}$	$380^{\pm 64}$	$64.04^{\pm 10.8}$
−0.001	1	$78.42^{\pm 0.63}$	$282^{\pm 105}$	$68.43^{\pm 6.50}$
−0.003	1	$\mathbf{78.26^{\pm 0.06}}$	$\mathbf{99^{\pm 36}}$	$\mathbf{76.49^{\pm 0.48}}$
−0.01	1	$78.10^{\pm 0.14}$	$391^{\pm 21}$	$75.83^{\pm 1.12}$
−0.03	1	$70.26^{\pm 1.46}$	$197^{\pm 2}$	$65.80^{\pm 1.41}$
−0.001	1.001	$78.45^{\pm 0.53}$	$283^{\pm 82}$	$65.03^{\pm 8.54}$
−0.003	1.003	$\mathbf{78.25^{\pm 0.14}}$	$\mathbf{119^{\pm 17}}$	$\mathbf{76.37^{\pm 0.45}}$

Table 5: The impact of clipped softmax hyperparameters on ViT-S/16 (trained for 150 epochs).

C Experimental details

C.1 BERT

Fine-tuning on MNLI dataset We use pre-trained checkpoint BERT-base-uncased (109M parameters) from HuggingFace repository. We follow standard fine-tuning practices from [14] and [62] Each data sequence is tokenized and truncated to the maximum sequence length of 128. Shorter sequences are padded to the same length of 128 using a special `[PAD]` token. We fine-tune for 3 epochs using Adam [28] with no weight decay. The learning rate is initially set to its maximum value and is linearly decayed to zero by the end of fine-tuning. We use a batch size of 16 and a maximum learning rate of $2 \cdot 10^{-5}$.

Pre-training from scratch We follow closely the pre-training procedure from [14]. We concatenate, tokenize, and split the training set into sequences of length 128 (to speed up training and

experimentation, we do not fine-tune on longer sequences of 512). We use the masked language modeling objective with the probability of masking $p = 0.15$. We train with a batch size of 256 sequences for 10^6 steps, using AdamW optimizer [37] with the maximum learning rate of 10^{-4}, learning rate warm up over the first 10^4 steps, following by a linear decay to zero by the end of training. We use L2 weight decay of 0.01, L2 gradient norm clipping of 1.0, and dropout probability of 0.1 on all layers. We also use FP16 mixed-precision from HuggingFace Accelerate library [19].

C.2 OPT pre-training

To speed up experimentation, we train OPT-125m sized model on the concatenation of Wikipedia and BookCorpus (same as BERT pre-training). We train with a batch size of 48 and 4 gradient accumulation steps (i.e., the effective batch size of 192), so that we can perform pre-training on a single A100 80GB GPU. We concatenate, tokenize, and split the training set into sequences of length 512 and train for 125000 steps (500000 forward passes).

We use the rest of the hyper-parameters and follow pre-training practices from [70] and [62]. We initialize weights using a normal distribution with zero mean and a standard deviation of 0.006. All bias terms are initialized to zero. We use AdamW optimizer with $(\beta_1, \beta_2) = (0.9, 0.95)$. We use the linear learning rate schedule, warming up from 0 to the maximum value of[1] $4 \cdot 10^{-4}$ over the first 2000 steps, following by a linear decay to zero by the end of training. We use L2 weight decay of 0.1, L2 gradient norm clipping of 1.0, and dropout probability of 0.1 on all layers. We also use FP16 mixed-precision from HuggingFace Accelerate library [19].

C.3 ViT pre-training

We use the model definition for ViT-S/16 and the training pipeline from PyTorch Image models library [61].

All training is done on resolution 224×224 and 16×16 patches. For data augmentation, we use RandAugment [10], Mixup [69], CutMix yun2019cutmix, random image cropping [53], horizontal flip, label smoothing $\varepsilon = 0.1$, color jitter 0.4, and random (between bilinear and bicubic) interpolation during training.

We train with a batch size of 512 for 300 epochs steps, using AdamW optimizer and the L2 weight decay of 0.03. We use the cosine learning rate schedule, warming up from 10^{-6} to the maximum value of 10^{-3} over the first 20 epochs, followed by a LR decay by a factor of 10 every 30 epochs, until it reaches the minimum value of 10^{-5}.

C.4 Quantization settings

Weights In all cases, we use symmetric uniform quantization of weights. We use min-max weight quantization for all models except the OPT model, for which we found the MSE estimator to perform better in all cases.

Activations We adopt *static range estimation* approach, which determines quantization parameters for the network by passing a few batches of calibration data through the model before inference. Specifically, we use a running min-max estimator [31], which uses an exponential moving average of the min and max over multiple batches. In all cases, we use running min-max with 0.9 momentum over 16 batches randomly sampled from respective training sets.

For OPT model, we also experiment with using 99.99% and 99.999% percentiles instead of actual min and max. We select the best configuration for each experiment (including baseline), based on the model performance. In almost all cases, we found that setting activation quantization ranges using 99.999% percentiles gives the lowest W8A8 perplexity.

D Compute cost

We compare the runtime of our proposed methods in Table 8. As we can see, the clipped softmax is only marginally more expensive compared to using the vanilla softmax attention. The gated attention

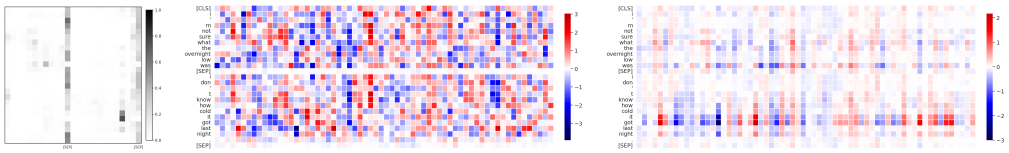
[1] We found this value to perform better compared to the value of $6 \cdot 10^{-4}$, listed in the paper.

Model	Vanilla	Clipped softmax	Gated attention (Linear / MLP)
BERT	$92.8^{\pm 1.2}$	$93.6^{\pm 0.8}$	97.7 / 119.1
OPT	$53.6^{\pm 0.4}$	$54.4^{\pm 0.4}$	55.7 / 64.7
ViT	$101.8^{\pm 0.3}$	$104.0^{\pm 0.7}$	110.8 / 122.9

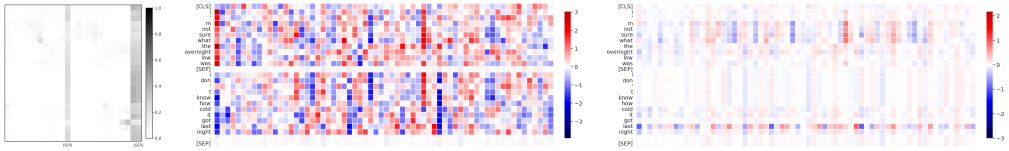
Table 6: An overview of the runtime of the proposed methods, compared to the vanilla pre-training, measured in hours on Nvidia-A100 GPUs.

using the linear $\boldsymbol{G}$ adds the compute overhead between 3% and 8%, depending on the model. We found that adding weight decay on LayerNorm $\boldsymbol{\gamma}$ for OPT and adding the LayerNorm after the patch embeddings for ViT had a negligible effect on the runtime.

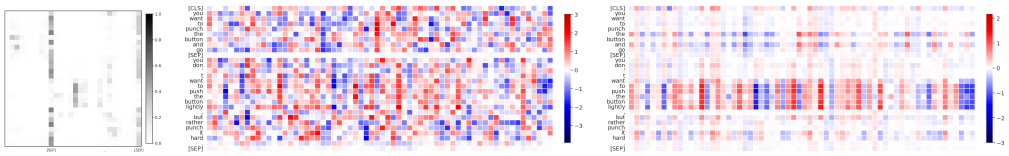
We estimated that the compute cost of producing the main results in the paper is about 320 GPU days (on A100) and the total cost of the project (including preliminary experiments and ablation studies) to be about 1400 GPU days.



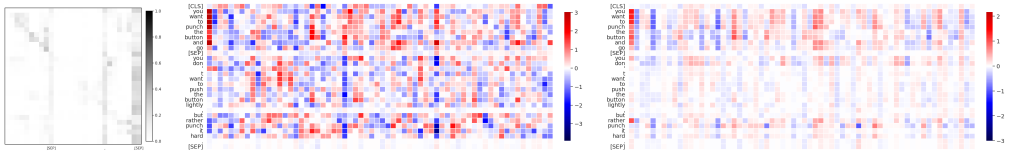
(a) Attention layer #10, data sequence #16



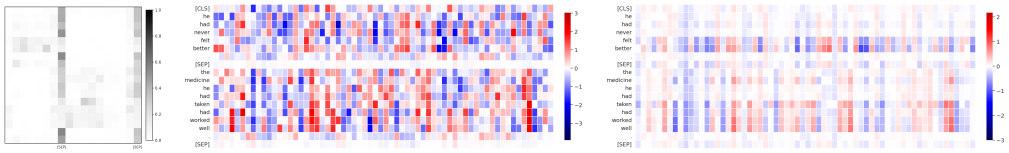
(b) Attention layer #11, data sequence #16



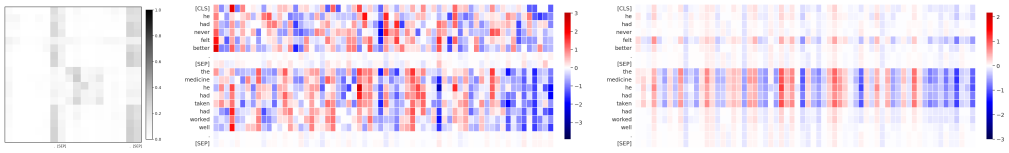
(c) Attention layer #10, data sequence #21



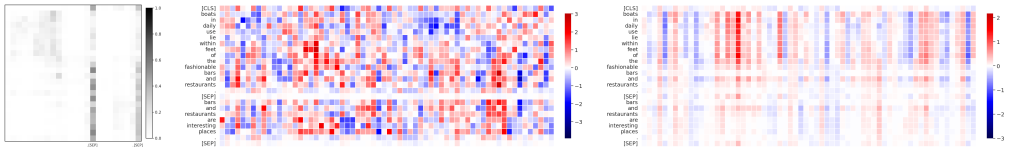
(d) Attention layer #11, data sequence #21



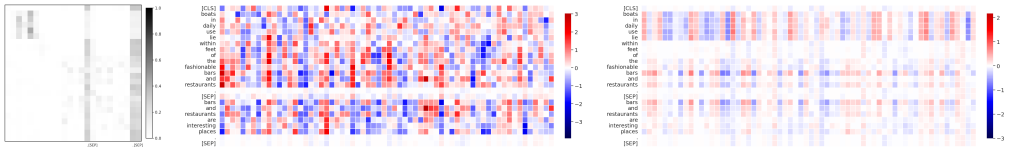
(e) Attention layer #10, data sequence #61



(f) Attention layer #11, data sequence #61

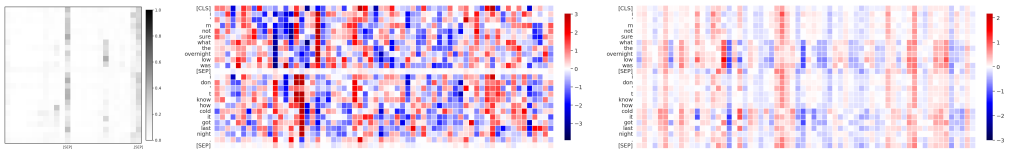


(g) Attention layer #10, data sequence #88

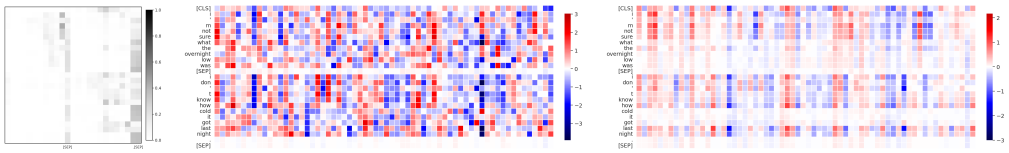


(h) Attention layer #11, data sequence #88

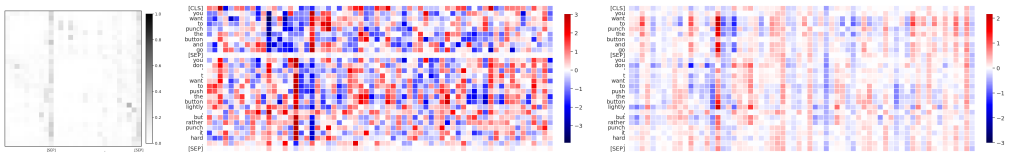
Figure 2: Visualization of the self-attention patterns (attention probabilities, values, and their product in left, middle and right columns, respectively) in attention head #3 ($\leftrightarrow$ channel dim #180) for BERT-base, computed on several random data sequences from MNLI-m validation set.



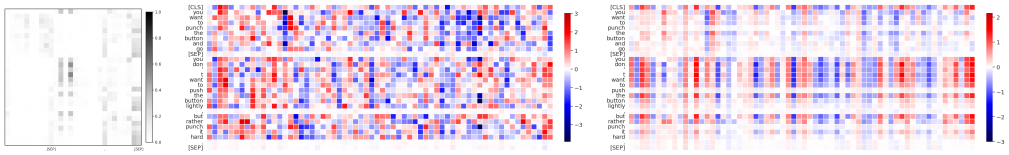
(a) Attention layer #10, data sequence #16



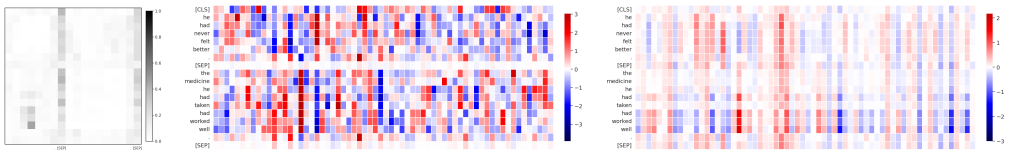
(b) Attention layer #11, data sequence #16



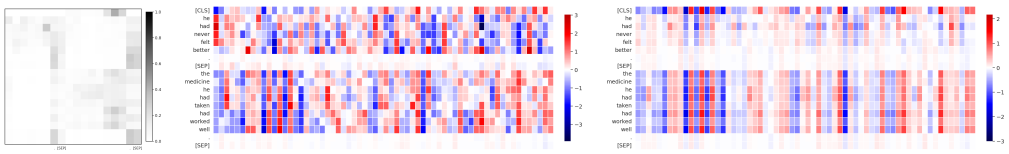
(c) Attention layer #10, data sequence #21



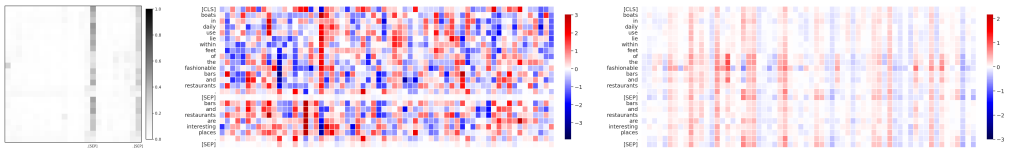
(d) Attention layer #11, data sequence #21



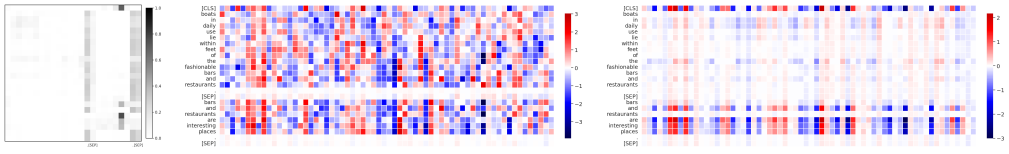
(e) Attention layer #10, data sequence #61



(f) Attention layer #11, data sequence #61

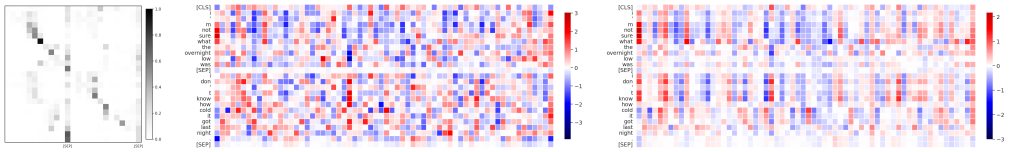


(g) Attention layer #10, data sequence #88

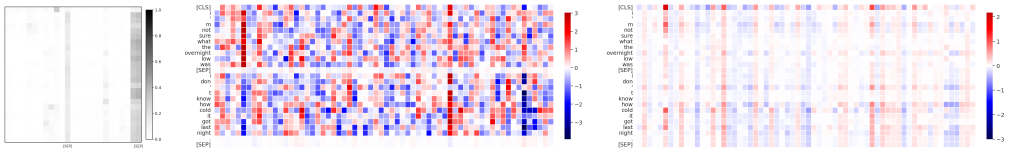


(h) Attention layer #11, data sequence #88

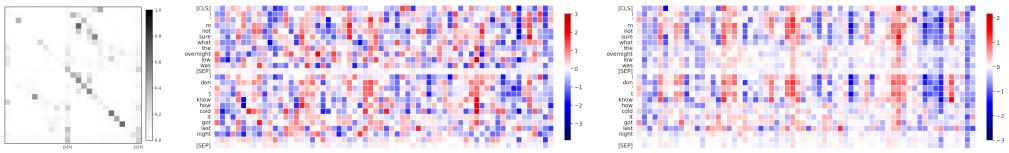
Figure 3: Visualization of the self-attention patterns (attention probabilities, values, and their product in left, middle and right columns, respectively) in attention head #12 ($\leftrightarrow$ channel dim #720) for BERT-base, computed on several random data sequences from MNLI-m validation set.



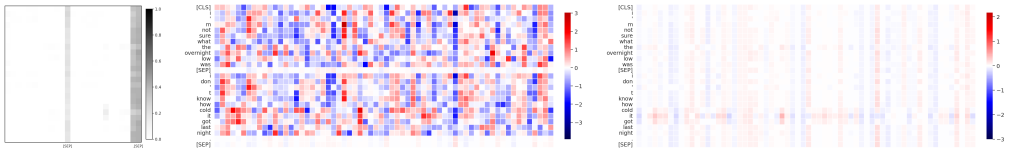
(a) Attention layer #10, attention head #1



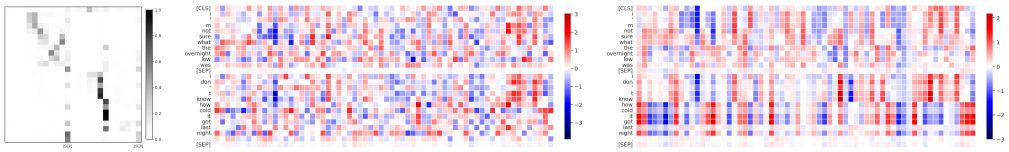
(b) Attention layer #11, attention head #1



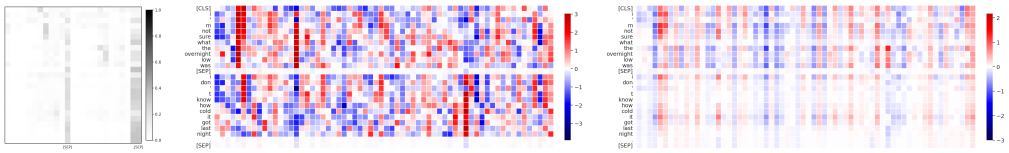
(c) Attention layer #10, attention head #7



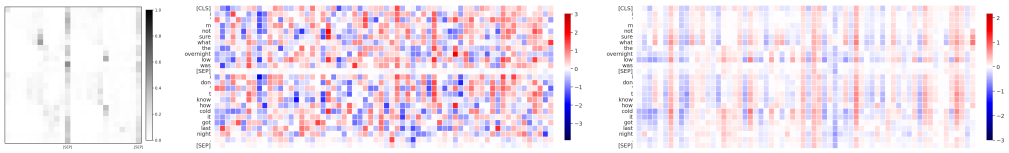
(d) Attention layer #11, attention head #7



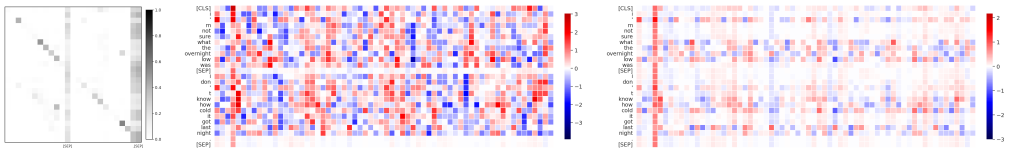
(e) Attention layer #10, attention head #8



(f) Attention layer #11, attention head #8

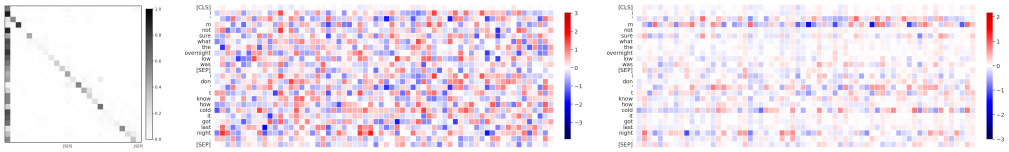


(g) Attention layer #10, attention head #10

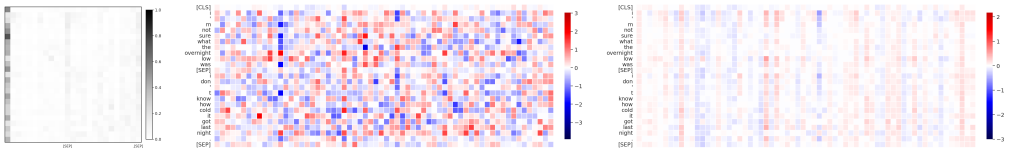


(h) Attention layer #11, attention head #10

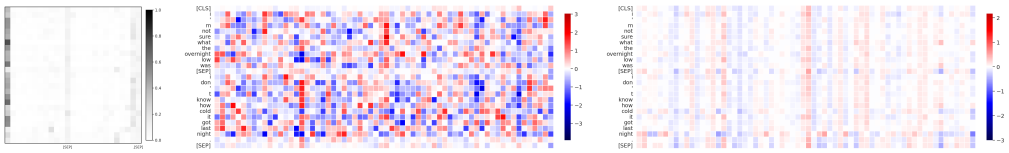
Figure 4: Visualization of the self-attention patterns (attention probabilities, values, and their product in left, middle and right columns, respectively) in attention heads that are not associated with the strong outliers for BERT-base, computed on data sequences #16 from MNLI-m validation set.



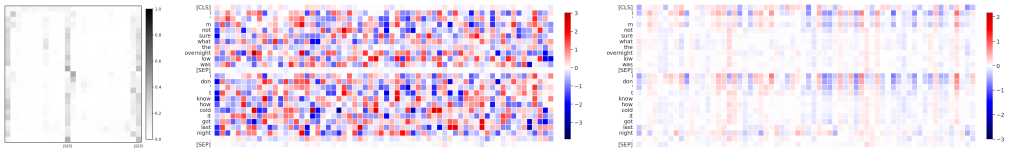
(a) Attention layer #1



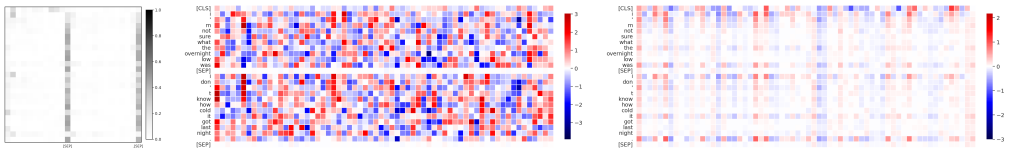
(b) Attention layer #2



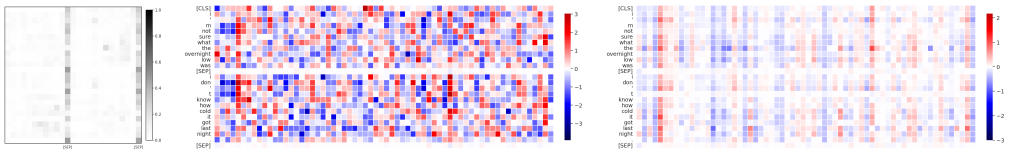
(c) Attention layer #3



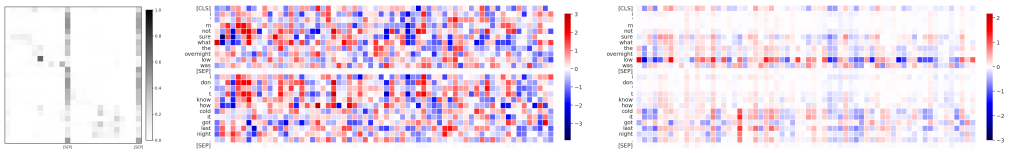
(d) Attention layer #4



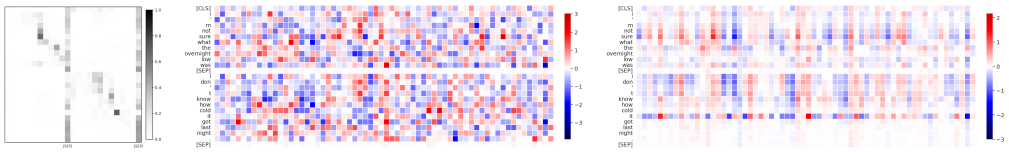
(e) Attention layer #5



(f) Attention layer #6



(g) Attention layer #7



(h) Attention layer #8

Figure 5: Visualization of the self-attention patterns (attention probabilities, values, and their product in left, middle and right columns, respectively) in attention head #3 ($\leftrightarrow$ channel dim #180) and the first eight layers of BERT-base, computed on data sequences #16 from MNLI-m validation set.

(a) (b) (c) (d) (e)

Figure 6: A summary of our outlier analysis for ViT demonstrated on a random subset from ImageNet validation set. (a) An input image. (b) Outliers in the output of layer #10. (c) Cumulative attention weight spent on every patch (attention probabilities matrix summed over rows) in the attention head #1, in the next layer #11. (d) A corresponding matrix of attention probabilities. (e) An average magnitude of values ($\boldsymbol{V}$) for outlier and non-outlier patches.

(a) (b) (c) (d) (e)

Figure 7: A summary of our outlier analysis for ViT demonstrated on a random subset from ImageNet validation set. (a) An input image. (b) Outliers in the output of layer #11. (c) Cumulative attention weight spent on every patch (attention probabilities matrix summed over rows) in the attention head #1, in the next layer #12. (d) A corresponding matrix of attention probabilities. (e) An average magnitude of values ($\boldsymbol{V}$) for outlier and non-outlier patches.