

# Softmax Bias Correction for Quantized Generative Models

Nilesh Prasad Pandey, Marios Fournarakis, Chirag Patel, Markus Nagel  
Qualcomm AI Research\*

{nileshpr, mfournar, cpatel, markusn}@qti.qualcomm.com

## Abstract

*Post-training quantization (PTQ) is the go-to compression technique for large generative models, such as stable diffusion or large language models. PTQ methods commonly keep the softmax activation in higher precision as it has been shown to be very sensitive to quantization noise. However, this can lead to a significant runtime and power overhead during inference on resource-constraint edge devices. In this work, we investigate the source of the softmax sensitivity to quantization and show that the quantization operation leads to a large bias in the softmax output, causing accuracy degradation. To overcome this issue, we propose an off-line bias correction technique that improves the quantizability of softmax without additional compute during deployment, as it can be readily absorbed into the quantization parameters. We demonstrate the effectiveness of our method on stable diffusion v1.5 and 125M-size OPT language model, achieving significant accuracy improvement for 8-bit quantized softmax.*

## 1. Introduction

The increasing prevalence of large generative neural networks, such as stable diffusion [4, 13, 24], ChatGPT, and OPT [33], has revolutionized the fields of computer vision and natural language processing. These models exhibit exceptional capabilities in generating realistic images and human-like text. However, deploying them on edge devices is challenging due to their size and computational demands. To address this issue, quantization has emerged as the most promising technique to optimize model deployment on resource-constrained devices, with a plethora of work emerging for both vision [14, 3, 18] and language models [19, 8, 6].

Post-training quantization (PTQ) is the go-to method

---

\*Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc

for quantizing such models because accessing original training data and pipelines can be difficult, and training them requires vast computing resources. However, activation quantization remains challenging because certain layers, such as the softmax in transformers, are particularly sensitive to quantization. This issue is even more pronounced in diffusion models due to the iterative nature of the denoising process leading to error accumulation. For this reason, it is common practice to keep the softmax unquantized or in higher precision leading to significant latency overhead, especially in networks with larger sequence lengths [29].

In this work, we systematically investigate the source of the softmax sensitivity to quantization and show that quantization operation leads to a large bias degrading accuracy. We introduce a hardware-friendly bias correction that acts as an offset at the softmax output, which can be absorbed into the quantization parameters. Despite its simplicity, our method significantly improves the SQNR and perplexity scores for diffusion [4] and OPT [33] language models, respectively, with 8-bit quantized softmax.

## 2. Background

### 2.1. Related Work

Quantization is one of the most effective methods available for reducing latency and power consumption in neural network inference. This is achieved not only thanks to reduced model size but because fixed-point operations are more efficient than their floating-point counterparts. In this work, we focus on post-training quantization (PTQ), which takes a pre-trained FP32 network and converts it directly into a fixed-point network without the need for the original training pipeline [20, 21, 15]. These methods require either no data or only a small calibration dataset and are easier to use compared to quantization-aware training (QAT) [5, 11, 10, 23]. For more details on neural network quantization, we refer the reader to [9, 22].

As the success of language models has increased concurrently with their size, a lot of recent work has focused on quantizing these models [27, 32, 27, 7]. While few methods have emerged to address the issue of outliers in the output of transformers [1, 2, 30], our work is complementary as we focus on quantizing the attention weights.

Similarly, while recent work on the quantizing diffusion models [17, 12, 26] have discussed various problems and methods to overcome quantization challenges, most of these methods keep sensitive activations, such as softmax, in higher-precision. However, softmax can be the biggest latency bottleneck due to its inefficient execution in hardware [29]. Our work is orthogonal to existing methods as we focus on improving the quantizability of softmax layers to lower bits.

## 2.2. Motivation

Softmax accounts for a significant fraction of the total runtime of transformers accounting for up to 40% for sequence lengths larger than 2048 [29]. As a result, keeping softmax in low precision can accelerate inference by reducing the size of the look-up tables required to estimate exponential functions. As modern diffusion model, e.g. stable diffusion v1.5<sup>1</sup> reach sequence lengths of 4096, low-bit softmax is imperative if we want to achieve competitive on-device performance.

However, when quantizing the softmax in stable diffusion to 8 bits, we observe a considerable deviation in the generated images compared to the floating-point model (see columns FP32 and W8A16-SM8 in figure 2). On the contrary, when keeping softmax at 16 bits (column W8A16 in figure 2), the generated image matches that of the floating-point model very closely.

To confirm our hypothesis that the softmax layers in the diffusion process are particularly sensitive to quantization, we perform the following sensitivity analysis: we quantize individual attention tensors to 8 bits in the denoising U-Net while keeping the rest of the network in FP32 and measure the signal-to-quantization noise ratio (SQNR) between the quantized and full-precision at the end of the denoising process. We use a calibration set  $\mathcal{X}$  of 400 input latents sampled uniformly across all time steps and report the mean SQNR in dB in table 1. We calculate the SQNR using the following formula:

$$\text{SQNR}_{\text{dB}} = 10 \log \mathbb{E}_{\mathbf{x}} \left[ \frac{\|\phi(\mathbf{x})\|_2^2}{\|q(\phi(\mathbf{x})) - \phi(\mathbf{x})\|_2^2} \right], \quad (1)$$

where  $\mathbf{x} \in \mathcal{X}$ ,  $\phi(\cdot)$  is the output of the denoising U-Net, and  $\|\cdot\|_2$  is the Forbenious norm.

<sup>1</sup><https://github.com/runwayml/stable-diffusion>

Activation in 8 bits	SQNR(↑)
Query (Q)	32.36
Key (K)	29.77
Value (V)	26.58
Attention score (softmax input)	28.09
Softmax output	3.24

Table 1: Quantization sensitivity analysis for attention layers in the denoising U-Net of stable diffusion. We quantize each activation to 8 bits while keeping the rest of the network unquantized, and report mean SQNR(↑): the higher, the better.

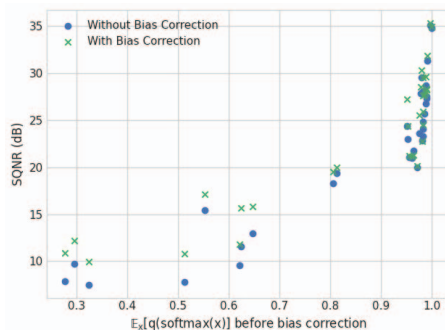


Figure 1: x-axis: sum of 8-bit quantized softmax vectors before bias correction while keeping rest of the network in FP32  $\mathbb{E}_{\mathbf{x}} [q(\text{softmax}(\mathbf{x}))]$ ; y-axis: SQNR between full-precision and quantized UNet outputs after final diffusion step.

We can see from table 1 that quantizing the softmax output leads to an 8-fold degradation in SQNR compared to the second most sensitive activation, the value tensor (V).

## 2.3. Quantized softmax is biased

Why is the softmax output so sensitive to quantization? Having a closer look at the values of the quantized softmax, we found that up to 99% of the values are rounded to zero. As all these values are rounded down, the resulting quantization error is *biased*, and the softmax probabilities are not correctly normalized anymore, which can degrade the model’s performance. In the scatter plot of figure 1, we see that many quantized softmax outputs do not add up to 1.0. In fact, the expected sum of the softmax output over the calibration set can be as low as 0.3. We also observe a high correlation between quantization bias and degradation of the denoising process: the larger the gap from the expected softmax output (1.0), the lower the SQNR at the U-Net output.

## 3. Quantized activation bias correction

In the previous section, we experimentally established that quantizing the softmax in the transformer can

lead to highly-biased outputs causing significant distortion in stable diffusion’s output. In this section, we outline a simple but effective method for correcting this bias and improving performance.

We define quantization bias as the systematic discrepancy between quantized and unquantized activation vector  $\mathbf{y}$ :

$$\beta(\mathbf{y}; \mathcal{T}) = \mathbb{E}[\mathcal{T}\mathbf{y}] - \mathbb{E}[q(\mathcal{T}\mathbf{y})], \quad (2)$$

where  $\mathcal{T}$  is the transformation function acting on  $\mathbf{y}$ , and  $q(\cdot)$  is the quantization function. The transformation  $\mathcal{T}$  could be the identity or a simple linear transformation, such as a reduction along a certain axis. We can now *correct* for this bias by adding back to the quantized activation  $\mathbf{y}_q = q(\mathcal{T}\mathbf{y})$ , such that

$$\mathbb{E}[\mathcal{T}\mathbf{y}_q + \beta(\mathbf{y}; \mathcal{T})] = \mathbb{E}[\mathcal{T}\mathbf{y}]. \quad (3)$$

In practice, we calculate an empirical estimate of the bias,  $\hat{\beta}$ , using the available calibration data.

### 3.1. Softmax bias correction

In the case of softmax activations, we know in advance that its output is normalized and should thus sum to 1.0. Using the notation of equation (2), the transformation  $\mathcal{T}$  is an inner product with the vector of ones along the normalization dimension:  $\mathbb{E}[\mathbf{1}^\top \mathbf{y}] = 1$ . In transformers, the input to the softmax layer is typically three-dimensional  $\mathbf{X} \in \mathbb{R}^{n_{\text{heads}} \times n_{\text{seq}} \times n_{\text{seq}}}$  and the softmax is applied across the last dimension. Depending on the capabilities of the hardware available, we could have a *per-tensor* or *per attention-head* correction factor, which would require reducing the output of the softmax output accordingly. For example, the *per-tensor* correction factor is calculated by:

$$\beta = \frac{1}{n_{\text{seq}}} - \frac{\mathbb{E}_{\mathbf{x}} \left[ \sum_i^{n_{\text{heads}}} \sum_j^{n_{\text{seq}}} \sum_k^{n_{\text{seq}}} \mathbf{Y}_{i,j,k} \right]}{n_{\text{heads}} n_{\text{seq}}^2}, \quad (4)$$

where  $\mathbf{Y} = q(\text{softmax}(\mathbf{X}))$  and  $\beta$  is added elementwise to the whole quantized output  $\mathbf{Y}$ . In later sections (cf. Sec 4.1), we perform an ablation study for bias correction granularities.

### 3.2. Absorbing bias correction

An important benefit of our bias correction method is that it can be easily absorbed into the offset of *asymmetric* quantization. For a  $b$  bitwidth uniform quantizer with scale  $s$  and zero-point  $z$ , asymmetric quantization is defined as:

$$\begin{aligned} y_q &= q(y; s, z, b) \\ &= s \cdot \left[ \text{clamp} \left( \left\lfloor \frac{y}{s} \right\rfloor + z, 0, 2^b - 1 \right) - z \right] \\ &= s \cdot y_{\text{int}} - c, \end{aligned} \quad (5)$$

Type of correction	SD (SQNR $\uparrow$ )
None	3.17
Per-tensor	5.77
Per attention-head	6.05
Time-step aware, per-tensor	5.93
Time-step aware, per attention-head	6.06

Table 2: Granularity ablation study for bias correction: we quantize the softmax to 8 bits, keeping the rest of the network in FP32. We report SQNR for stable diffusion (SD) ( $\uparrow$ ): the higher, the better

where  $c = s \cdot z$  is a floating-point offset of the quantization grid, as the scale  $s$  is typically a floating-point number [15]. For bias correction, we only have to absorb the correction factor into the offset,  $c' = s \cdot z - \beta$ , while keeping everything else the same. Given that activations are most commonly asymmetrically quantized [1, 30, 22, 2], our bias correction leads to no additional compute.

## 4. Experiments

In this section, we demonstrate the advantages of softmax bias correction. We perform experiments on stable diffusion v1.5<sup>2</sup> and extend our analysis to a transformer-based language model. We experiment with the 125M-sized variant of OPT [33] pre-trained using the causal language modeling (CLM) objective. We use a validation pipeline for HuggingFace libraries [31, 16] and evaluate on Wikipedia validation set<sup>2</sup>. We report SQNR between the full-precision and quantized U-Net output for stable diffusion (the higher, the better), and CLM perplexity for OPT (the lower, the better).

We use PyTorch v1.11 and the AI Model Efficiency Toolkit (AIMET)<sup>3</sup> [28] to quantize the models to desired bitwidths. We implement per-tensor symmetric quantization for weights and asymmetric quantization for activations.

### 4.1. Granularity of bias correction

As mentioned in section 3.1, depending on the target hardware, we can apply bias correction at different granularities of attention tensor, *e.g.* *per-attention head* or *per-tensor*. Due to the iterative nature of the denoising process in diffusion models, activation distributions in the U-Net are time-step dependent, hence motivating

<sup>2</sup><https://github.com/huggingface/diffusers>

<sup>2</sup>Specifically, we use the English subset of Wiki-40b, <https://huggingface.co/datasets/wiki40b>, that contains cleaned-up text of English Wikipedia and training/validation splits.

<sup>3</sup>AIMET is a product of Qualcomm Innovation Center, Inc., available on GitHub at <https://github.com/quic/aimet>

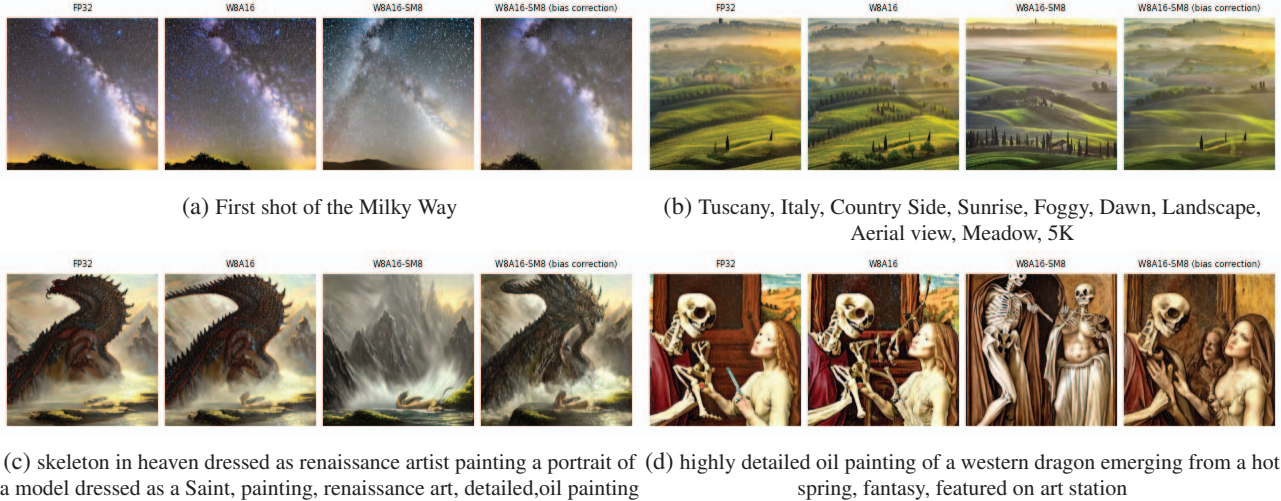


Figure 2: Visual comparison between FP32, W8A16, W8A16 with softmax quantized to 8 bits without and with per attention head based bias correction generated using 20 diffusion steps on test prompts from LAION Aesthetics dataset [25].

us to use the time-step as an additional axis of granularity to perform *time-step-aware* bias correction. We compare the results from the different schemes in table 2.

We observe that the per-attention head correction scheme performs on par or better than all other schemes, making it a favorable choice for on-device deployment due to its minimal computational overhead compared to its time-aware counterpart.

## 4.2. Main Results

We extend our analysis to include the 125M-size OPT language model, and we report results using per-attention head bias correction in table 3. We quantize softmax to 8 bits (SM8) and keep the rest of the network at either full-precision (FP32) or 8-bit weights and 16-bit activations (W8A16). With bias correction, we achieve over 2.7dB improvement for stable diffusion and roughly 4.8 improvement in perplexity for OPT in both quantization settings. In the case of diffusion, we also demonstrate the improvement visually in figure 2, by showing the generated images of the quantized diffusion with and without bias correction. As we can see, the generated image with bias correction very closely resembles the full precision output.

## 5. Conclusions

In this work, we investigated a common prevailing issue of softmax sensitivity to quantization in the case of generative models. To understand the source of the softmax sensitivity to quantization, we analyzed

Configuration	SD (SQNR $\uparrow$ )	OPT (ppl $\downarrow$ )
FP32 baseline	-	27.73
FP32-SM8	3.17	34.98
FP32-SM8 + bias correction	6.05	30.19
W8A16 baseline	9.66	27.77
W8A16-SM8	3.05	35.11
W8A16-SM8 + bias correction	5.76	30.24

Table 3: Per-attention bias correction for stable diffusion (SD) and 125M OPT model for different quantization configurations (W8A16 & FP32), while keeping the softmax to 8 bits (SM8). We report SQNR for stable diffusion (SD) and CLM perplexity for OPT. ( $\uparrow$ ): the higher, the better; ( $\downarrow$ ): the lower, the better.

the softmax distributions and showed that the quantization operation led to a significant bias in the softmax output degrading the performance of generative models when represented in lower precision. To overcome this issue, we proposed a simple yet effective hardware-friendly offset correction to improve the quantizability of softmax layers to lower bits, which is key to achieving competitive on-device performance, especially for transformer-based networks with longer sequence lengths. We demonstrated the effectiveness of our method on stable diffusion v1.5 and 125M-size OPT language model, achieving over 2.7dB improvement for stable diffusion and roughly 4.8 improvement in perplexity for OPT respectively for the 8-bit weights and 16-bit activations (W8A16) quantization setting.

## References

- [1] Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Understanding and overcoming the challenges of efficient transformer quantization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7947–7969, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics.
- [2] Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Quantizable transformers: Removing outliers by helping attention heads do nothing, 2023.
- [3] Yu-Hui Chen, Raman Sarokin, Juhyun Lee, Jiuqiang Tang, Chuo-Ling Chang, Andrei Kulik, and Matthias Grundmann. Speed is all you need: On-device acceleration of large diffusion models via gpu-aware optimizations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4650–4654, 2023.
- [4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- [5] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019.
- [6] Elias Frantar and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.
- [7] Elias Frantar, Saleh Ashkboos, Torsten Hoeftler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers, 2023.
- [8] Elias Frantar, Saleh Ashkboos, Torsten Hoeftler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023.
- [9] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. *CoRR*, abs/2103.13630, 2021.
- [10] Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazhen Lin, Fengwei Yu, and Junjie Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. *International Conference on Computer Vision (ICCV)*, 2019.
- [11] Tiantian Han, Dong Li, Ji Liu, Lu Tian, and Yi Shan. Improving low-precision network quantization via bin regularization. In *International Conference on Computer Vision (ICCV)*, 2021.
- [12] Yefei He, Luping Liu, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Ptdq: Accurate post-training quantization for diffusion models. *arXiv preprint arXiv:2305.10657*, 2023.
- [13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [14] Jilei Hou and Ziad Asghar. World’s first on-device demonstration of stable diffusion on an android phone. 2023.
- [15] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018.
- [16] Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matuszière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander M. Rush, and Thomas Wolf. Datasets: A community library for natural language processing, 2021.
- [17] Xiuyu Li, Long Lian, Yijiang Liu, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. *arXiv preprint arXiv:2302.04304*, 2023.
- [18] Yanyu Li, Huan Wang, Qing Jin, Ju Hu, Pavlo Chemerys, Yun Fu, Yanzhi Wang, Sergey Tulyakov, and Jian Ren. Snapfusion: Text-to-image diffusion model on mobile devices within two seconds. *arXiv preprint arXiv:2306.00980*, 2023.
- [19] Yijiang Liu, Huanrui Yang, Zhen Dong, Kurt Keutzer, Li Du, and Shanghang Zhang. Noisyquant: Noisy bias-enhanced post-training activation quantization for vision transformers, 2023.
- [20] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, pages 7197–7206. PMLR, 2020.
- [21] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1325–1334, 2019.
- [22] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. A white paper on neural network quantization. *arXiv preprint arXiv:2106.08295*, 2021.
- [23] Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. Overcoming oscillations in quantization-aware training. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 16318–16330. PMLR, 17–23 Jul 2022.

- [24] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [25] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.
- [26] Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1972–1981, 2023.
- [27] Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Q-bert: Hessian based ultra low precision quantization of bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8815–8821, 2020.
- [28] Sangeetha Siddegowda, Marios Fournarakis, Markus Nagel, Tijmen Blankevoort, Chirag Patel, and Abhijit Khobare. Neural network quantization with ai model efficiency toolkit (aimet). *arXiv preprint arXiv:2201.08442*, 2022.
- [29] Jacob R Stevens, Rangharajan Venkatesan, Steve Dai, Brucek Khailany, and Anand Raghunathan. Softermax: Hardware/software co-design of an efficient softmax for transformers. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 469–474. IEEE, 2021.
- [30] Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. Outlier suppression+: Accurate quantization of large language models by equivalent and optimal shifting and scaling. *arXiv preprint arXiv:2304.09145*, 2023.
- [31] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, Oct. 2020. Association for Computational Linguistics.
- [32] Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. Q8bert: Quantized 8bit bert. *arXiv preprint arXiv:1910.06188*, 2019.
- [33] 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. Opt: Open pre-trained transformer language models, 2022.