NoisyQuant: Noisy Bias-Enhanced Post-Training Activation Quantization for Vision Transformers
(Supplementary Materials)

Yijiang Liu[*1], Huanrui Yang[*2], Zhen Dong[2], Kurt Keutzer[2], Li Du[1✉], Shanghang Zhang[3✉]
[1]Nanjing University, [2]University of California, Berkeley
[3]National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University
liuyijiang@smail.nju.edu.cn, {huanrui, zhendong, keutzer}@berkeley.edu
ldu@nju.edu.cn, shanghang@pku.edu.cn

This document provides additional visualizations and experimental results to support the main paper. We demonstrate results of quantization error on model output in Appendix A, visualize prediction scores for ImageNet classes in Appendix B, illustrate more histogram examples of the input and output activation distributions on different transformer layers in Appendix C, discuss memory and computation overhead in Appendix D, and show additional experimental results in Appendix E.

A. Quantization error of model output

In this section, we show the comparison of the output logits between EasyQuant [10] and NoisyQuant with 6-bit ViT [1], DeiT [8] and Swin [5] models. Here NoisyQuant is implemented on top of EasyQuant with the proposed noisy bias enhancement. We go through the whole ImageNet validation set and calculate the mean-square error of model output logits on each quantized model compared to the pretrained floating-point counterpart. As shown in Tab. 1, NoisyQuant achieves quantization error reduction on all model outputs, especially for ViT-S (17%) and Swin-T (16%) models.

B. Visualization of model output

Following Appendix A, we visualize model output in Fig. 1 to give further perspectives on how the reduced quantization error achieved by NoisyQuant improved final accuracy. Specifically, we plot prediction logits produced by the floating-point (red), EasyQuant (gray), and NoisyQuant (green) models on the 1000 ImageNet [7] classes, respectively. The highest logits are marked with stars, where the location of the red star corresponds to the ground truth class. With less quantization error, NoisyQuant logits closely match that of the floating-point model, thus achieving better performance than EasyQuant.

Table 1. **Quantization error of model output.** Models are quantized by EasyQuant and NoisyQuant with the W6A6 setting.

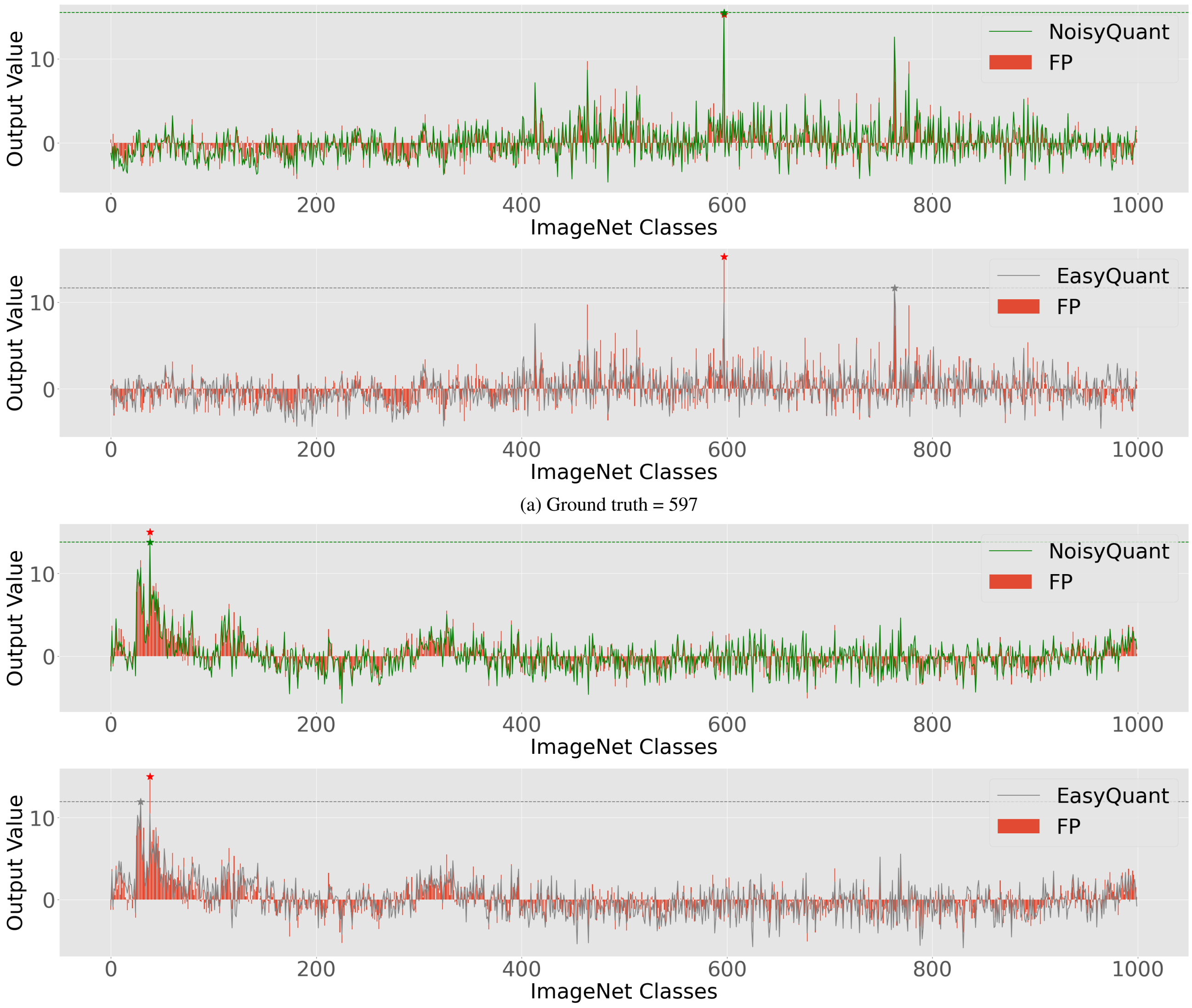
Model	EasyQuant [10]	NoisyQuant	Reduction
ViT-S	1.0400	0.8583	0.1818 **(17%)**
ViT-B	0.6365	0.5982	0.0383 **(6%)**
ViT-B*	0.6956	0.6360	0.0596 **(9%)**
DeiT-S	0.3270	0.2934	0.0335 **(10%)**
DeitT-B	0.2869	0.2584	0.0284 **(10%)**
DeiT-B*	0.1984	0.1760	0.0224 **(11%)**
Swin-T	0.0913	0.0765	0.0148 **(16%)**
Swin-S	0.0296	0.0289	0.0007 **(2%)**
Swin-B	0.0505	0.0457	0.0047 **(9%)**
Swin-B*	0.0412	0.0399	0.0013 **(3%)**

C. Additional input and output activation histogram

In this section, we present more histogram examples of layer input and output as previously described in Sec. 4.2 of the main paper. We briefly illustrate the pipeline of EasyQuant and NoisyQuant in Fig. 2. The top-left sub-figure refers to input activation X, and EasyQuant follows the gray arrow while NoisyQuant follows the blue. NoisyQuant utilizes the proper-selected noisy bias N to refine the input before quantization (shown in the bottom-left sub-figure). The output histograms are shown in the right sub-figures, and we point out the mismatch caused by EasyQuant with the orange arrow.

As we have emphasized in the main paper, transformer layers produce sophisticated activation distributions. Fig. 2 gives more examples from different transformer layers. Fig. 2a and Fig. 2b show fc2 layers in ViT-S and DeiT-S

* Equal contribution.
✉ Corresponding Author.



(a) Ground truth = 597

(b) Ground truth = 38

Figure 1. Model output of the floating-point (red), EasyQuant (green), and NoisyQuant (gray) model. The floating-point and NoisyQuant models give correct predictions (red/green star) while EasyQuant gives wrong prediction (gray star).

which takes GELU [2] activations as input; the asymmetric and heavy-tailed input activation distribution makes a negative impact on the layer output produced by EasyQuant. Instead, NoisyQuant refines the distribution to achieve a better match in the quantized layer output. Fig. 2c gives an example of the downsample layer in Swin models which as well enjoys the noisy bias enhancement.

D. Memory and computation overhead

Memory overhead. In practice, for weights $W \in \mathbb{R}^{k\times m}$ and activations $X \in \mathbb{R}^{m\times n}$, we follow the standard implementation to set bias $B \in \mathbb{R}^{k\times 1}$ and sample noise $N \in \mathbb{R}^{m\times 1}$, so the denoising bias $B' = B - q_W(W)N$ is also $\mathbb{R}^{k\times 1}$, where $q_W(\cdot)$ is the quantizer. The sum follows the broadcasting rule. Storing N brings minimal overhead, for instance, DeiT-B* has 86.9M params, with only 0.06M (0.07%) for storing the noise.

Computation overhead. The matrix multiplication, i.e., WX, dominates the computation of ViT linear layers, requiring $10^3\times$ more MAC than the number of adds in $X + N$ and bias. So the cost of FP32 add is negligible (<0.4%) to that of INT8 layer. Further, N and B' can be INT16 rather than FP32, enabling integer-only inference and reducing the cost of add to <0.03%. We observe no accuracy differences in using INT16 or FP32 for N and B' in our experiments. We estimate the energy cost with 0.23pJ/Int8-

Table 2. Performance with smaller calibration set.

Size	**W/A**	**ViT-S**	**ViT-B**	**ViT-B***	**DeiT-S**	**DeiT-B**	**DeiT-B***	**Swin-T**	**Swin-S**	**Swin-B**	**Swin-B***
32	6/6	76.81	81.89	82.81	76.30	79.71	81.19	79.97	82.74	84.55	85.90
128	6/6	76.87	81.97	82.86	76.47	80.20	81.24	80.13	82.68	84.44	86.00
1024	6/6	76.86	81.90	83.00	76.37	79.77	81.40	80.01	82.78	84.57	85.90

Table 3. Comparing to reparameterization.

Model	W/A	Reparam.	**NoisyQuant**
ViT-S	6/6	76.66	**78.90** ± 0.06
DeiT-B	6/6	81.03	**81.26** ± 0.04
Swin-S	6/6	82.46	**82.83** ± 0.04

MAC, 0.9pJ/FP32-Add, and 0.05pJ/Int16-Add following [3].

E. Additional experiments

Ablation study on calibration size. We follow [6]'s setting for calibration size 1024. Further experiments show that calibration size as low as 32 can still produce similar performance (see Tab. 2).

Additional baselines. Concurrent works [4, 9] introduce the reparameterization approach which reparameterizes LN layer to suppress outliers by scaling down activation values. NoisyQuant is orthogonal as we actively change the activation distribution being quantized without scaling. So NoisyQuant can be plugged in after reparameterization. We reproduce the reparameterization used in the two works and subsequently add NoisyQuant to show consistent improvement in Tab. 3.

Additional models. Beyond ViT, on ResMLP-24 with W6A6, NoisyQuant (76.71%) beat EasyQuant (76.48%) by 0.23%.

Performance at low bit. NoisyQuant outperforms EasyQuant by 1.06% and 4.60% respectively on 5-/4-bit Swin-T.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[2] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 2

[3] Mark Horowitz. 1.1 computing's energy problem (and what we can do about it). In *ISSCC*, 2014. 3

[4] Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. Repq-vit: Scale reparameterization for post-training quantization of vision transformers. *arXiv preprint arXiv:2212.08254*, 2022. 3

[5] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021. 1

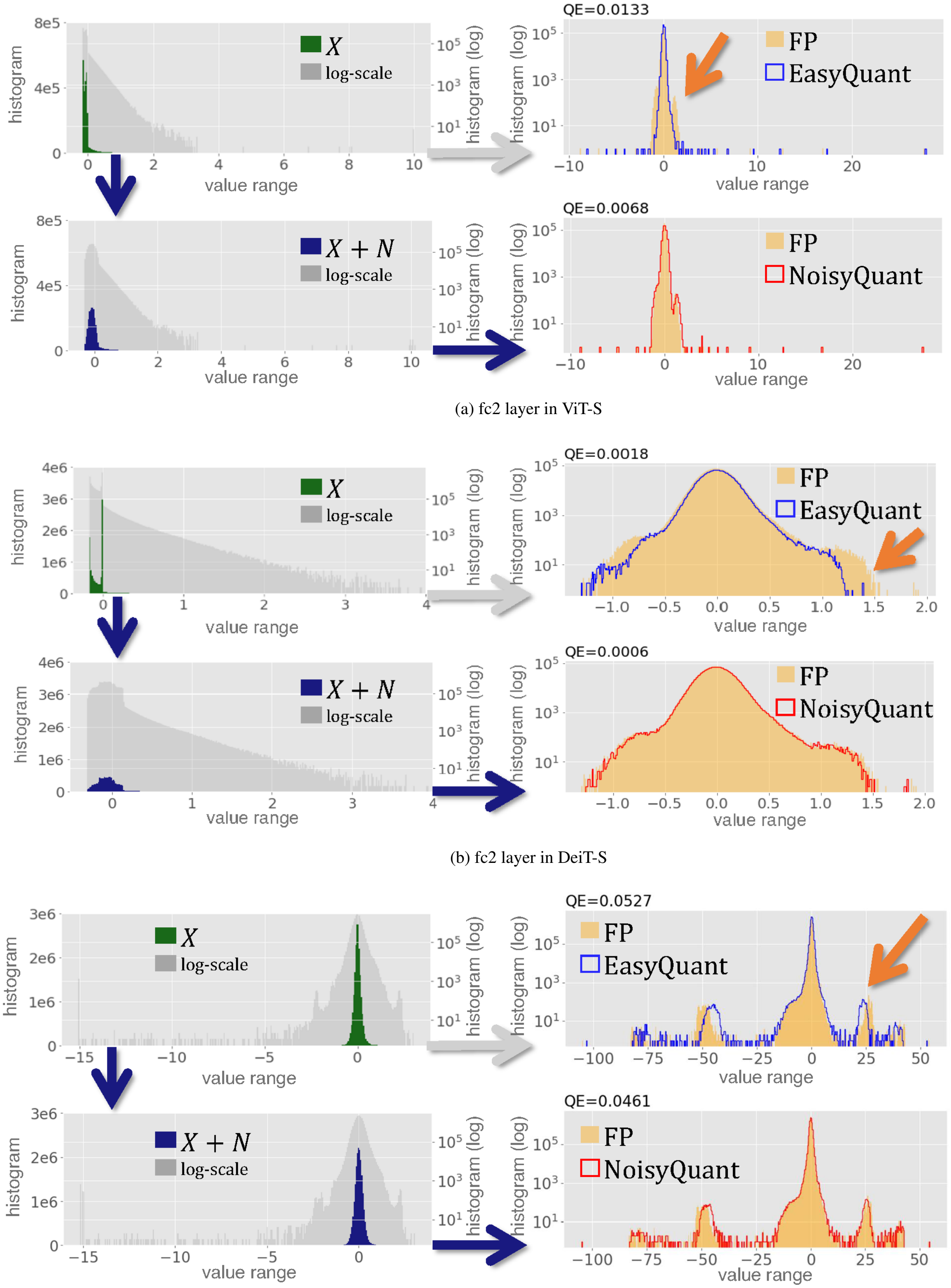
[6] Zhenhua Liu, Yunhe Wang, Kai Han, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. In *NeurIPS*, 2021. 3

[7] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015. 1

[8] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herv'e J'egou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 1

[9] Xiuying Wei, Yunchen Zhang, Xiangguo Zhang, Ruihao Gong, Shanghang Zhang, Qi Zhang, Fengwei Yu, and Xianglong Liu. Outlier suppression: Pushing the limit of low-bit transformer language models. *arXiv preprint arXiv:2209.13325*, 2022. 3

[10] Di Wu, Qi Tang, Yongle Zhao, Ming Zhang, Ying Fu, and Debing Zhang. Easyquant: Post-training quantization via scale optimization. *arXiv preprint arXiv:2006.16669*, 2020. 1



(a) fc2 layer in ViT-S

(b) fc2 layer in DeiT-S

(c) downsample layer in Swin-B

Figure 2. Input (left) and output (right) histogram on different layers.