QUANTDEMOIRE: QUANTIZATION WITH OUTLIER AWARE FOR IMAGE DEMOIRÉING
Supplementary Material

Zheng Chen[1]*, **Kewei Zhang**[1]*, **Xiaoyang Liu**[1], **Weihang Zhang**[2],
Mengfan Wang[2], **Yifan Fu**[2], **Yulun Zhang**[1]†
[1]Shanghai Jiao Tong University, [2]Central Media Technology Institute, Huawei

1 MORE ABLATION

Table 1: Ablation on the sampling rate in the activation. Evaluation is conducted on UDHM, 4-bit.

(a) Smooth sampling rate (γ_1).

Rate	10^{-2}	10^{-3}	10^{-4}
PSNR ↑	19.92	**20.92**	20.80
SSIM ↑	0.6941	**0.7570**	0.7505
LPIPS ↓	0.4141	**0.3171**	0.3221

(b) Bound sampling rate (γ_2).

Rate	10^{-2}	10^{-3}	10^{-4}
PSNR ↑	20.88	**20.92**	20.89
SSIM ↑	0.7558	**0.7570**	0.7474
LPIPS ↓	0.3183	**0.3171**	0.3388

For the smooth stage (γ_1), we ablate the sampling rate by varying γ_1 while fixing $\gamma_2 = 10^{-3}$. As shown in Tab. 1a, the setting that produces the best performance corresponds to $\gamma_1 = 10^{-3}$. Higher rates introduce excessive outliers in the smoothing process, whereas lower rates bias the preliminary estimate. A moderate sampling rate, therefore, provides the most stable and accurate range estimation.

For the bound stage (γ_2), we perform an analogous ablation by varying γ_2 while fixing $\gamma_1 = 10^{-3}$. Tab. 1b indicates that $\gamma_2 = 10^{-3}$ gives the best results. Larger rates lead to the accumulation of outliers during refinement, and smaller rates cause biased bounds. These results confirm that a moderate sampling rate also optimizes accuracy, reinforcing our overall design choice.

2 IMPLEMENTATION DETAILS OF COMPARISON METHODS

MinMax. We apply per-channel quantization to the weights and per-tensor quantization to the activations. During the calibration stage, we directly use the minimum and maximum activation values of the activation tensor as the quantization boundaries.

Percentile. We also use per-channel quantization for the weights and per-tensor quantization for the activations. Specifically, for both weights and activations, the 99.9%-th and 0.1%-th percentiles of all values are consistently used as the quantization boundaries.

2DQuant. We first employ the DOBI (Liu et al., 2024) method to search for quantizer parameters. Then we train the model using the quantization parameters obtained from DQC (Liu et al., 2024), with a training configuration of 100 epochs, a batch size of 4, and a learning rate of 10^{-2} . All other training settings remain consistent with those used in our proposed method.

SVDQuant. We first apply SmoothQuant (Xiao et al., 2023) to mitigate outliers in the activations. Next, we perform an SVD decomposition on the weights and compute both the low-rank and residual branches. To ensure fairness in our experiments and to prevent the low-rank branch from introducing an excessive number of additional parameters, we set the rank to 2.

3 VARIANTS OF SVDQUANT

We observed that SVDQuant suffers from a substantial performance drop when quantized to extremely low bit-widths (3 and 4 bits). To investigate this issue, we conducted further experiments with higher ranks. The results in Tab. 2 show that when the rank is increased to 8, SVDQuant achieves a significant performance improvement. However, when the rank is set to 8, although the performance improves, the relatively small parameter size of the convolutional layers in ESDNet leads SVDQuant to introduce an excessive number of additional parameters. Therefore, in our experiments, we set the rank of SVDQuant to 2 for comparative fairness (parameter).

*Equal contribution.

†Corresponding author: Yulun Zhang, yulun100@gmail.com

Table 2: Results of SVDQuant (rank = 2 and 8). Evaluation is conducted on UDHM, 3, and 4-bit.

Method	Bit(w/a)	Params (M)	Ops (G)	UHDM PSNR ↑	UHDM SSIM ↑	UHDM LPIPS ↓
SVDQuant (rank = 2)	4/4	1.08	3.76	14.68	0.5762	0.6559
SVDQuant (rank = 8)	4/4	2.10	8.48	18.69	0.7428	0.3615
QuantDemoire (ours)	4/4	**0.79**	**1.80**	**21.08**	**0.7626**	**0.3068**
SVDQuant (rank = 2)	3/3	0.90	3.34	14.83	0.4547	0.7289
SVDQuant (rank = 8)	3/3	2.01	6.80	17.20	0.6707	0.5033
QuantDemoire (ours)	3/3	**0.61**	**1.38**	**19.12**	**0.6839**	**0.4567**

Table 3: Compression ratios of Params and Ops at 8/6/4/3-bit. Ops are measured with an input size of $3\times224\times224$. Our QuantDemoire maintains efficiency and performance.

Method	Bit (w/a)	Params (M) (↓Ratio)	Ops (G) (↓Ratio)	UHDM PSNR ↑	UHDM SSIM ↑	UHDM LPIPS ↓
ESDNet (Yu et al., 2022)	32/32	5.93 (↓0%)	13.52 (↓0%)	22.12	0.7956	0.2551
MinMax (Jacob et al., 2018)	8/8	1.49 (↓74.90%)	3.38 (↓75.00%)	21.50	0.7727	0.2596
Percentile (Li et al., 2019)	8/8	1.49 (↓74.90%)	3.38 (↓75.00%)	19.38	0.7744	0.2784
2DQuant (Liu et al., 2024)	8/8	1.49 (↓74.90%)	3.38 (↓75.00%)	21.20	0.7827	0.2749
SVDQuant (Li et al., 2024)	8/8	1.82 (↓69.29%)	5.44 (↓59.75%)	21.80	0.7907	0.2580
QuantDemoire (ours)	8/8	1.53 (↓74.25%)	3.47 (↓74.31%)	22.00	0.7932	0.2555
MinMax (Jacob et al., 2018)	6/6	1.12 (↓81.14%)	2.54 (↓81.25%)	20.48	0.7648	0.2828
Percentile (Li et al., 2019)	6/6	1.12 (↓81.14%)	2.54 (↓81.25%)	18.44	0.7562	0.2957
2DQuant (Liu et al., 2024)	6/6	1.12 (↓81.14%)	2.54 (↓81.25%)	20.02	0.7595	0.2893
SVDQuant (Li et al., 2024)	6/6	1.45 (↓75.53%)	4.60 (↓65.97%)	20.86	0.7602	0.3015
QuantDemoire (ours)	6/6	1.16 (↓80.43%)	2.64 (↓80.49%)	21.61	0.7874	0.2572
MinMax (Jacob et al., 2018)	4/4	0.75 (↓87.38%)	1.69 (↓87.50%)	16.51	0.5255	0.6786
Percentile (Li et al., 2019)	4/4	0.75 (↓87.38%)	1.69 (↓87.50%)	16.85	0.6701	0.4639
2DQuant (Liu et al., 2024)	4/4	0.75 (↓87.38%)	1.69 (↓87.50%)	17.07	0.6288	0.5117
SVDQuant (Li et al., 2024)	4/4	1.08 (↓81.78%)	3.76 (↓72.19%)	14.68	0.5762	0.6559
QuantDemoire (ours)	4/4	0.79 (↓86.61%)	1.80 (↓86.68%)	21.08	0.7626	0.3068
MinMax (Jacob et al., 2018)	3/3	0.56 (↓90.51%)	1.27 (↓90.63%)	12.53	0.3802	0.8630
Percentile (Li et al., 2019)	3/3	0.56 (↓90.51%)	1.27 (↓90.63%)	14.79	0.5206	0.6869
2DQuant (Liu et al., 2024)	3/3	0.56 (↓90.51%)	1.27 (↓90.63%)	11.20	0.2465	0.8460
SVDQuant (Li et al., 2024)	3/3	0.90 (↓84.89%)	3.34 (↓75.30%)	14.83	0.4547	0.7289
QuantDemoire (ours)	3/3	0.61 (↓89.69%)	1.38 (↓89.78%)	19.12	0.6839	0.4567

4 More Compression Ratio

A more comprehensive comparison of compression ratios is presented in Tab. 3. Our method achieves performance close to that of the full-precision model, while maintaining a high compression ratio.

5 More Qualitative Results

Additional visual comparisons are presented in Figs. 1 and 2. Our method demonstrates clear advantages under the 3-, 4-, and 6-bit settings, as well as across all datasets.

6 More Distribution Visualizations

Additional distributions of weights and activations are presented in Figs. 3, 4, 5, and 6. It can be observed that the activations in most convolutional layers approximately follow either an exponential or a Gaussian distribution, while the weights in the majority of convolutional layers exhibit an approximately Gaussian distribution overall. This further proves our point.

7 Mathematical Details for the Sampling Analysis

In the main text, we introduced the sampling-based range estimation method. In this section, we provide mathematical proofs to support this approach.

7.1 A PROBABILITY-THEORETIC ANALYSIS

Assumptions and Conditions. In Sec. 6, we have already demonstrated that the activation distributions of most layers can be well approximated by a Gaussian distribution. For convenience, we assume that the activation tensor $\mathbf{X}$ consists of N independent random variables, each following a standard normal distribution. We then randomly sample $n = \gamma N$ of these variables to estimate the range of activations. Without loss of generality, let us set $N = 10^8$ and $\gamma = 10^{-3}$, which is consistent with the actual inputs and the experimental settings.

Distribution of the Maximum. Formally, let $X_1, \ldots, X_N \overset{iid}{\sim} \mathcal{N}(0, 1)$ and denote the maximum by $M_N = \max_{1 \le i \le N} X_i$. Its density is calculated as follows:

$$f_{M_N}(x) = N\phi(x)\Phi(x)^{N-1}, \tag{1}$$

where ϕ and Φ are the standard normal density and CDF, respectively. The expectation is then:

$$\mathbb{E}[M_N] = \int_{-\infty}^{\infty} xN\phi(x)\Phi(x)^{N-1}dx. \tag{2}$$

Similarly, the expectation maximum of the n randomly selected activation values is:

$$\mathbb{E}[M_n] = \int_{-\infty}^{\infty} xn\phi(x)\Phi(x)^{n-1}dx. \tag{3}$$

Since these two integrals do not admit closed-form solutions, we evaluated them numerically using Python, obtaining $\mathbb{E}[M_N] = 5.301$ and $\mathbb{E}[M_n] = 4.384$.

Moreover, it is straightforward to verify that $\Phi(\mathbb{E}[M_n) = 0.9999941$, which indicates that the expected maximum among the sampled elements corresponds to the 99.99941-th percentile of the entire activation population. In practice, however, the distribution of outliers is often observed to deviate even further from the main body of the activation values. This observation confirms that our method, while sampling only a vanishingly small fraction of elements, is able to discard extreme outliers while still providing an accurate estimate of the effective range of the main distribution.

8 PSEUDOCODE OF QUANTDEMOIRE

To provide a clearer description of the procedure of our method, we present the following pseudocode in Alg. 1 for the sampling-based activation quantizer.

REFERENCES

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *CVPR*, 2018.

Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. Svdquant: Absorbing outliers by low-rank components for 4-bit diffusion models. *arXiv preprint arXiv:2411.05007*, 2024.

Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. Fully quantized network for object detection. In *CVPR*, 2019.

Kai Liu, Haotong Qin, Yong Guo, Xin Yuan, Linghe Kong, Guihai Chen, and Yulun Zhang. 2dquant: Low-bit post-training quantization for image super-resolution. In *NeurIPS*, 2024.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *ICML*, 2023.

Xin Yu, Peng Dai, Wenbo Li, Lan Ma, Jiajun Shen, Jia Li, and Xiaojuan Qi. Towards efficient and scale-robust ultra-high-definition image demoiréing. In *ECCV*, 2022.

Shanxin Yuan, Radu Timofte, Gregory Slabaugh, Aleš Leonardis, Bolun Zheng, Xin Ye, Xiang Tian, Yaowu Chen, Xi Cheng, Zhenyong Fu, et al. Aim 2019 challenge on image demoireing: Methods and results. In *ICCVW*, 2019.

Algorithm 1: Pipeline of Sampling-Based Activation Quantizer

Initialization:
$model \leftarrow \mathrm{ESDNet(fp32)}$
$N \leftarrow$ number of batches in calibration dataset
for *convolution layers* L_i *in* $model$ **do**
  $W_i \leftarrow$ pretrained weight of L_i
  $C_i \leftarrow$ number of input channels ofL_i
  $lb_a^i \leftarrow 0,\ ub_a^i \leftarrow 0$
  $lb_w^i \leftarrow 0,\ ub_w^i \leftarrow 0$
  $\boldsymbol{s}^i[1:C_i] \leftarrow 0$
end
Calibration Stage 1:
for x *in calibration dataset* **do**
  $model(x)$
  for *convolution layers* L_i *in* $model$ **do**
    $\mathbf{A} \leftarrow$ latest activation tensor of L_i
    $\boldsymbol{s}^i[c] \leftarrow \max\left|\mathrm{Sample}_\gamma(\mathbf{A}_c)\right|, \quad \forall c = 1, \ldots, C_i$
  end
end
for *convolution layers* L_i *in* $model$ **do**
  $\boldsymbol{s}^i \leftarrow \boldsymbol{s}^i / N$
end
Calibration Stage 2:
for x *in calibration dataset* **do**
  $model(x)$
  for *convolution layers* L_i *in* $model$ **do**
    $\mathbf{A} \leftarrow$ latest activation tensor of L_i
    $lb_a^i \leftarrow lb_a^i + \min(\mathrm{Sample}_\gamma(\mathbf{A} \oslash \boldsymbol{s}^i))$
    $ub_a^i \leftarrow ub_a^i + \max(\mathrm{Sample}_\gamma(\mathbf{A} \oslash \boldsymbol{s}^i))$
  end
end
for *convolution layers* L_i *in* $model$ **do**
  $lb_a^i \leftarrow lb_a^i / N$
  $ub_a^i \leftarrow ub_a^i / N\ W_i \leftarrow W_i \oslash \boldsymbol{s}^i$
end

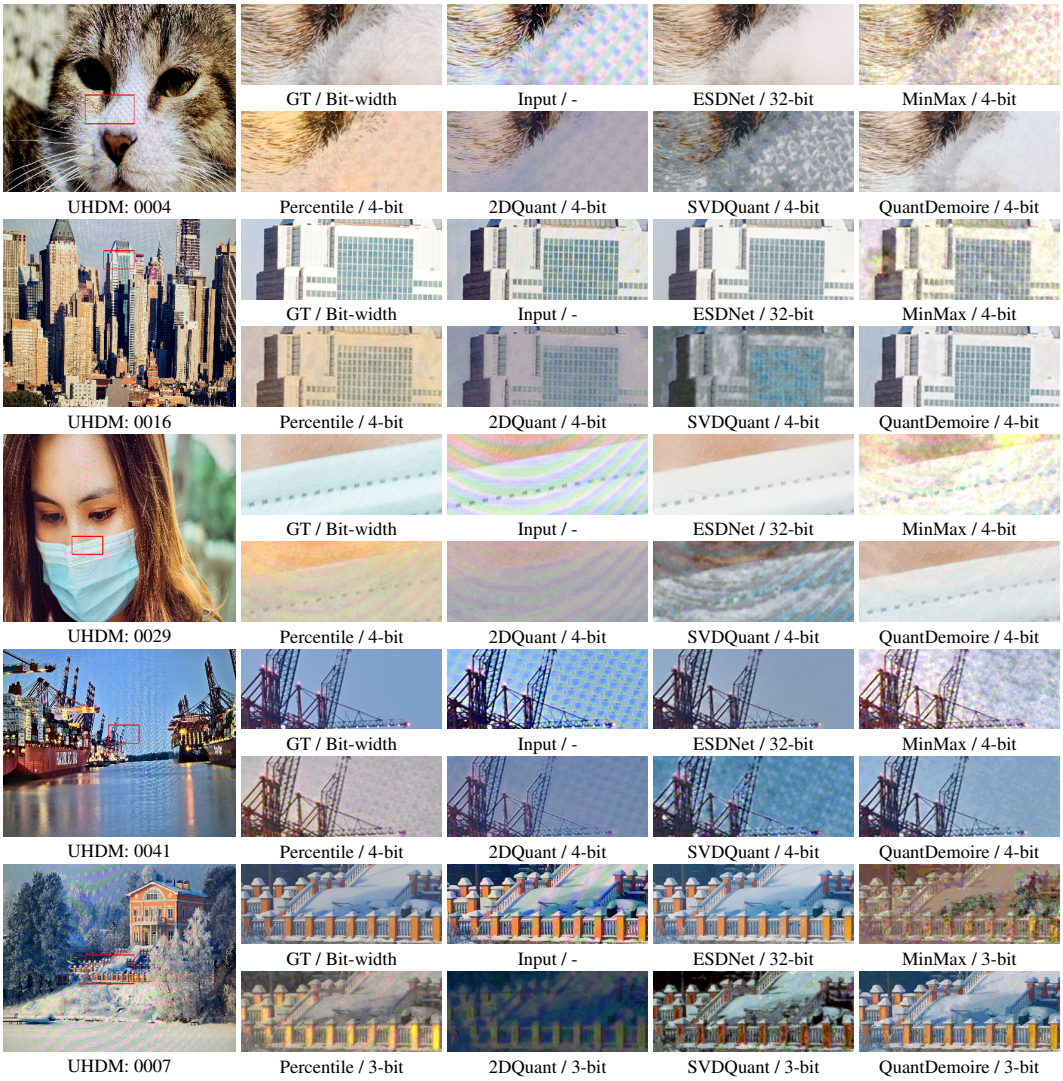

Figure 1: Visual comparison on the UHDM (Yu et al., 2022) dataset.

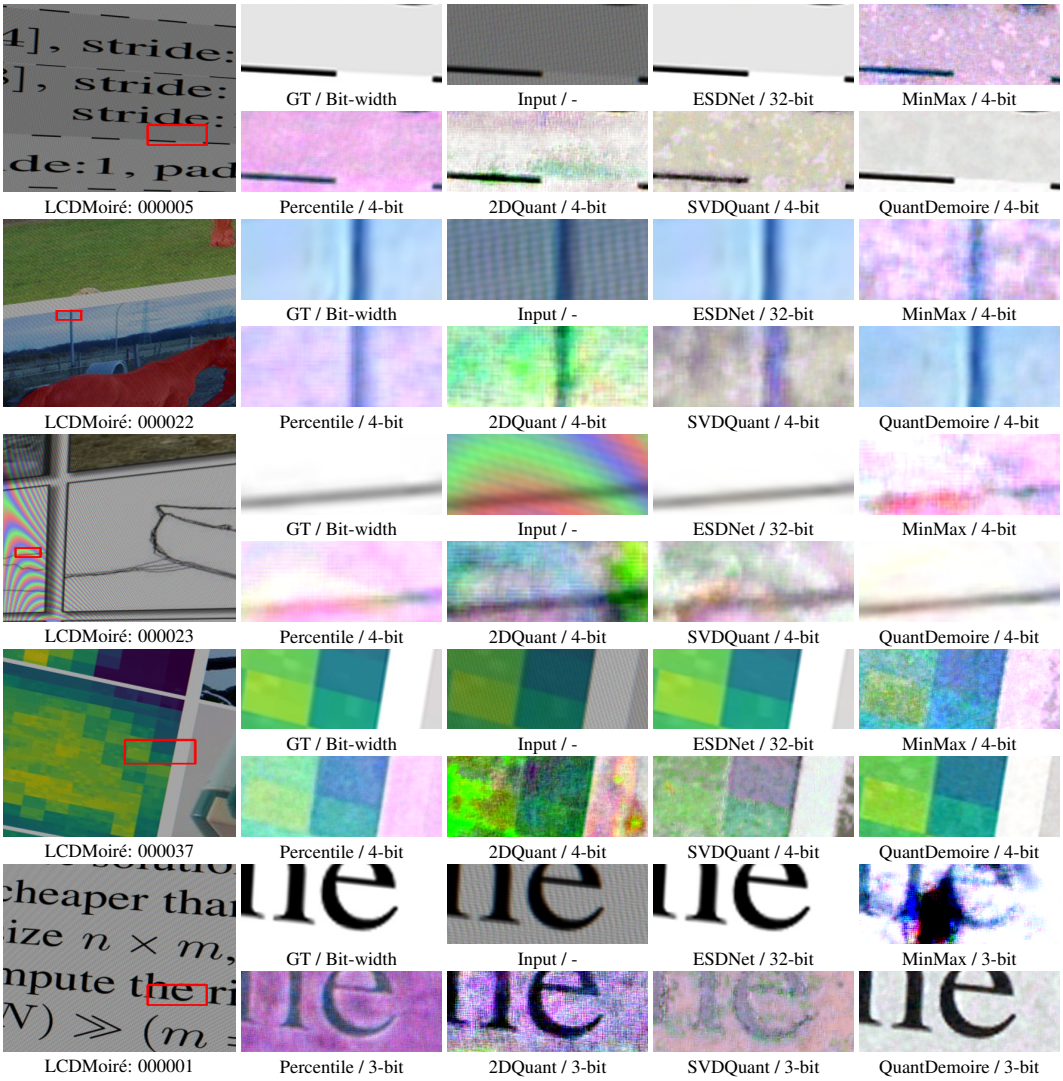

Figure 2: Visual comparison on the LCDMoiré (Yuan et al., 2019) dataset.

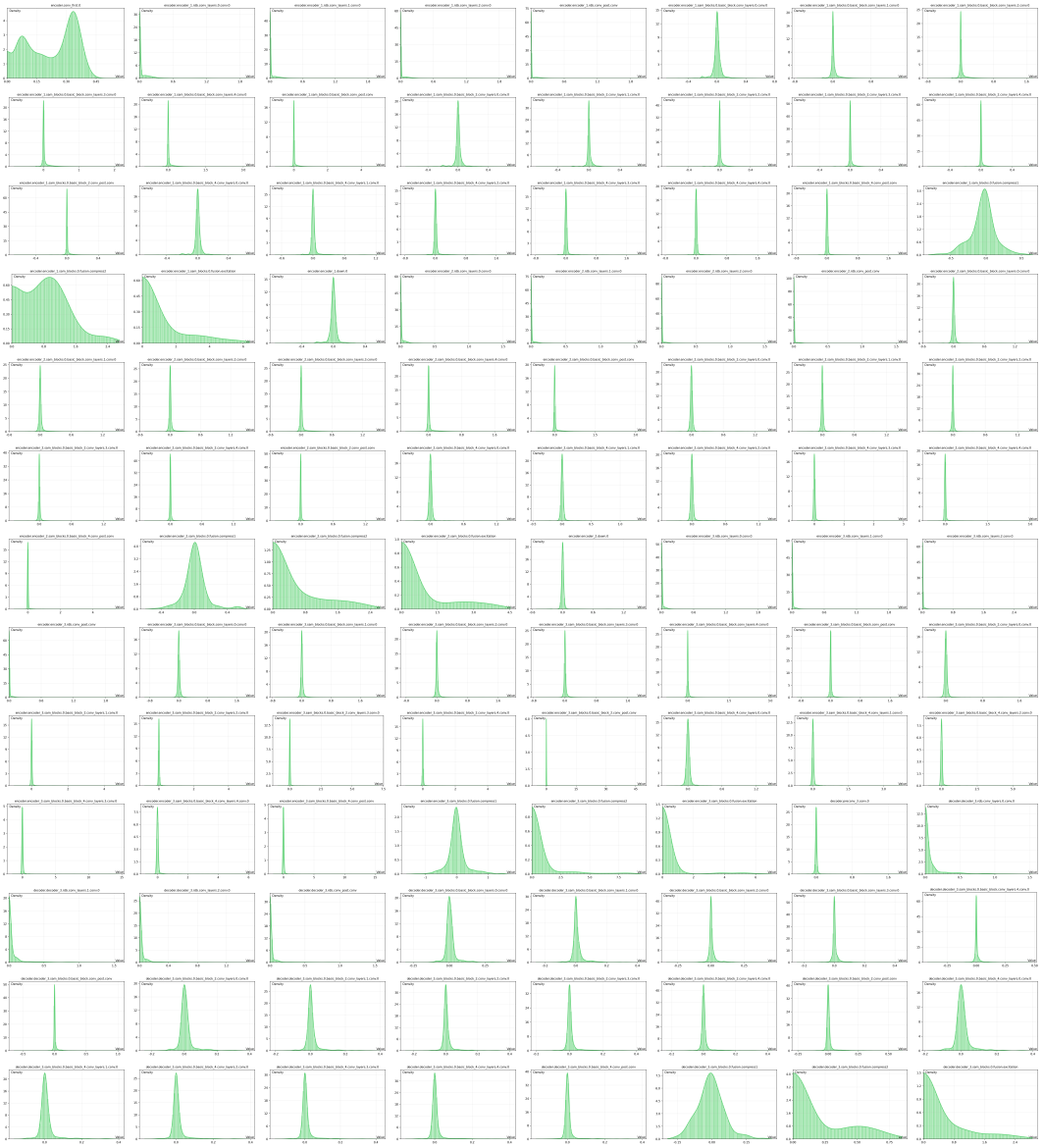

Figure 3: More distribution of activation (Part 1).

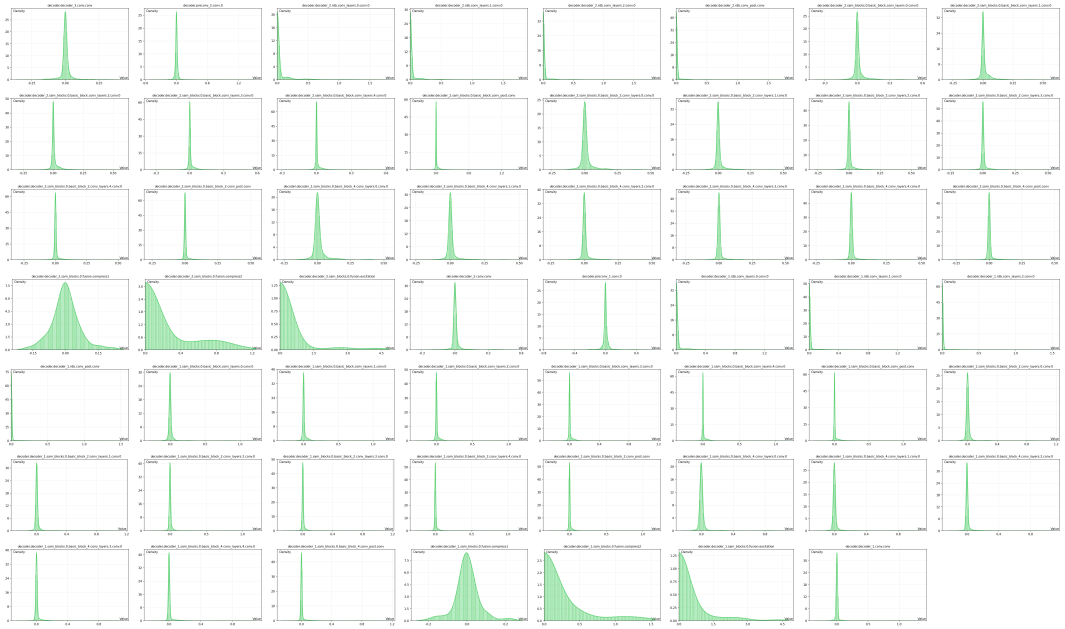

Figure 4: More distribution of activation (Part 2).

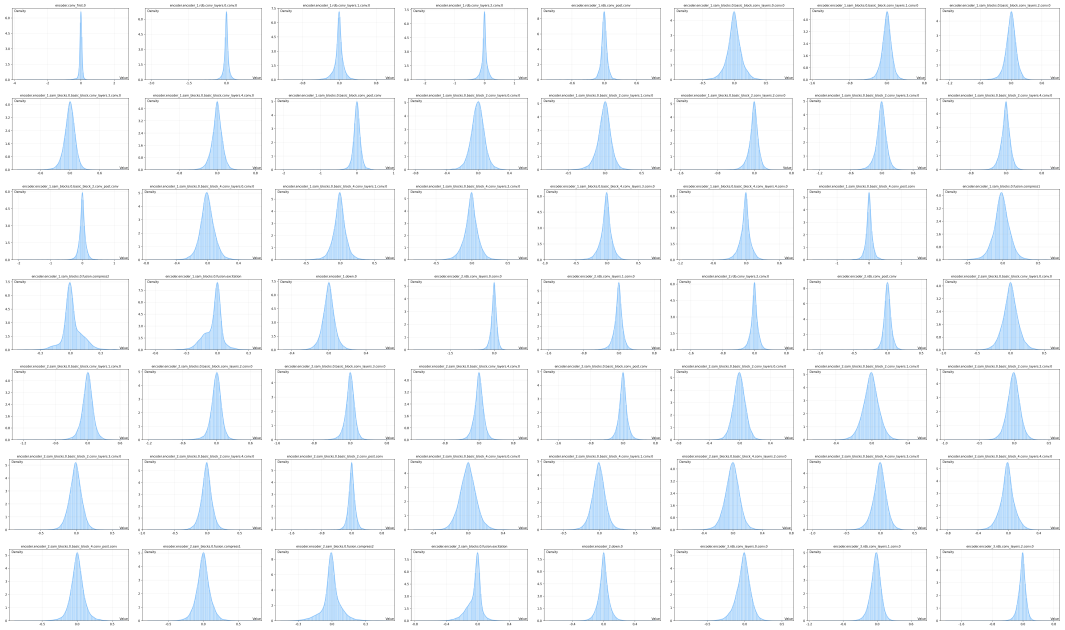

Figure 5: More distribution of weight (Part 1).

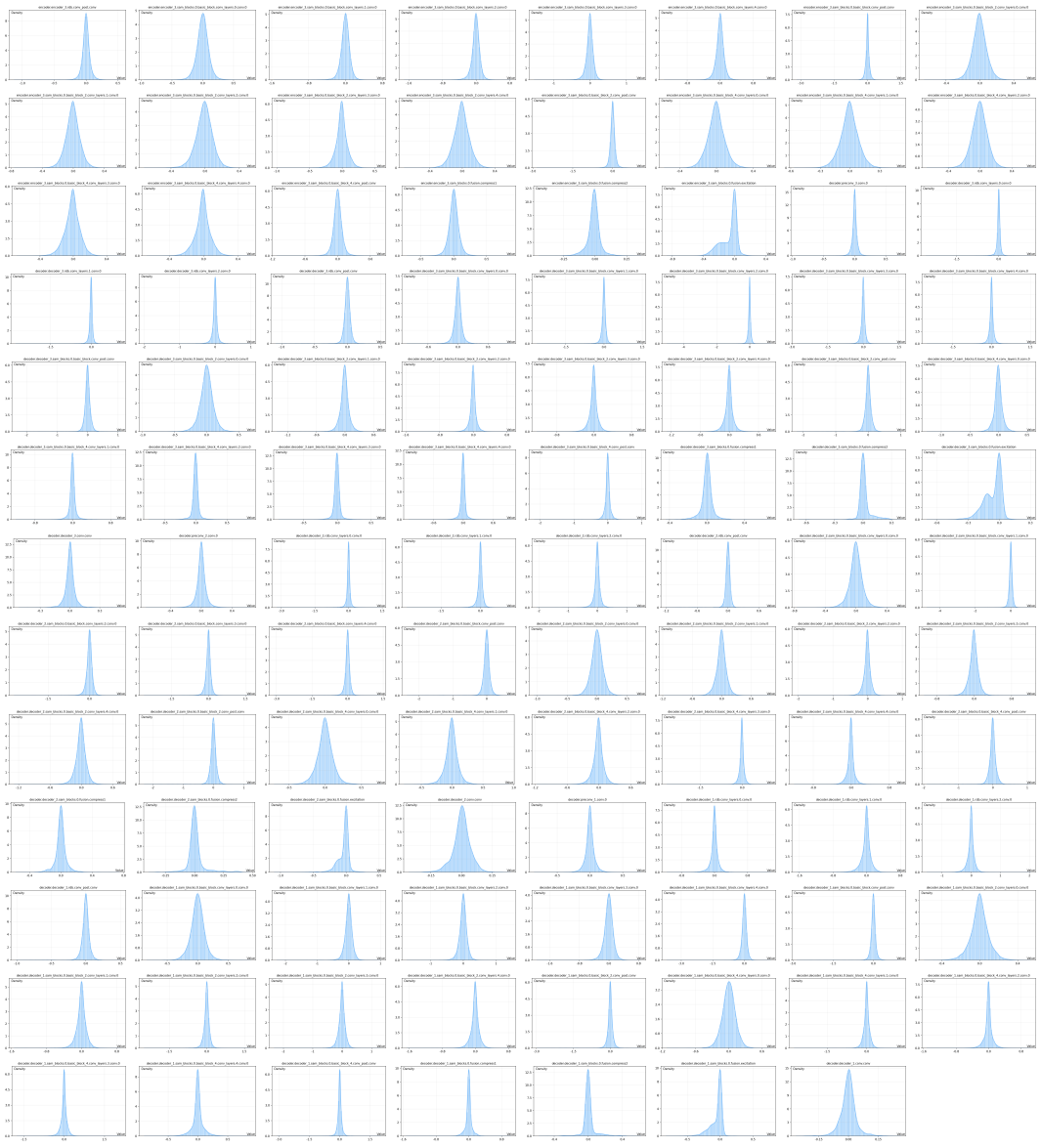

Figure 6: More distribution of weight (Part 2).