

# Appendix

## FraQAT: Quantization Aware Training with Fractional bits

Anonymous CVPR submission

Paper ID

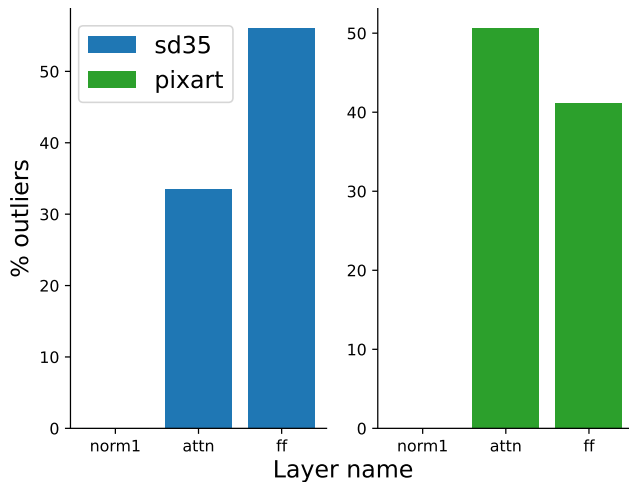


Figure 1. **Outliers:** outliers distribution for activations varies across models. SD3.5-M (left) experience most of its outliers right after Feed Forward layers, while for PixArt- $\Sigma$ , most outliers are in Attention layers.

Table 1. **Outlier analysis:** we optimize specific layers types while the rest of the model is frozen and quantized (w4a8). FID and CLIP-FID are computed on PixArt- $\Sigma$  [1] evaluation dataset.

Model	Layer	FID ↓	CLIP FID ↓
SD3.5-M	FF	<b>2.23</b>	0.23
	Attn	2.32	0.24
	TF	2.49	0.28
	All	2.54	<b>0.22</b>
Sana 600M	FF	2.18	0.17
	Attn	<b>2.10</b>	<b>0.16</b>
	TF	2.13	<b>0.16</b>
	All	2.17	0.19
PixArt- $\Sigma$	FF	5.34	1.55
	Attn	6.48	2.23
	TF	<b>4.40</b>	<b>1.13</b>
	All	4.48	1.07

### A. Experimental evaluation

#### A.1. Baselines

For state of the art baselines we rely on code released by authors<sup>12</sup> and use the default parameters. For all approaches we use pre-trained models with default resolution 512 × 512. Where needed we change the baselines configurations to use the same model.

#### A.2. Outlier analysis

Activation outliers disrupt the quantization process by introducing artifacts or biases. By analyzing these outliers across different models, we discover that different models produce outliers in different layers. For example, in SD3.5-M outliers emerge after Feed-Forward (FF) layers, while in PixArt- $\Sigma$  outliers arise from Attention (Attn) layers, see

<sup>1</sup>SVDQuant <https://github.com/mit-han-lab/deepcompressor>

<sup>2</sup>DiTAS <https://github.com/DZY122/DiTAS>

Figure 1. Through selectively training specific layers, we can reduce FraQAT’s computational demand while obtaining a deployable model. In this vein, we analyze the impact of selective training, i.e., we optimize only certain layers while the rest of the network is frozen and quantized (w4a8). In particular, we focus on attention layers (Attn), feed forward layers (FF), and transformer blocks (TF), and compare it with training the entire network (Full).

Quantitative results in Table 1 show that there is no clear winner – a layer type for all architecture. Different models take advantage from optimizing different layers. Nevertheless we recommend starting from quantizing Transformer Blocks (TF) as it reduces memory requirements, lowers computational demands, and addresses all outliers.

#### A.3. Hyper-parameters for QAT

We detail the various hyper parameters for all QAT experiments in Table 2. In all cases we rely on FuseAdam as optimizer and optimize for 25 epochs. All experiments run on

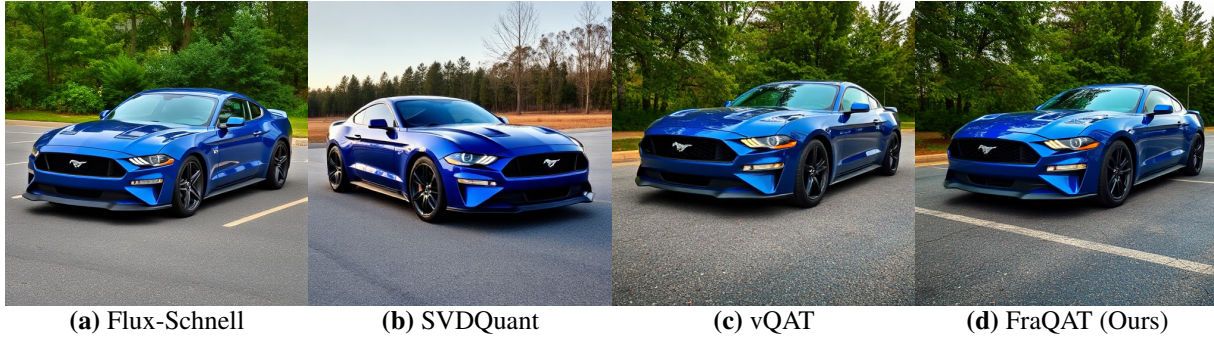


Figure 2. **Qualitative comparison:** FraQAT(d) generates images similar to the original model (a). Prompts are from MJHQ dataset [3].

AMD MI300X and are implemented using PyTorch<sup>3</sup>, Lightning<sup>4</sup>, torchao<sup>5</sup>, with seed 1234.

For all FraQAT experiments, we follow the schedule highlighted in Table 3.

Experiments with the configuration highlighted above take on average 192 GPUh for Sana, 576 GPUh for PixArt-Σ, 1008 GPUh for SD3.5-Medium.

#### A.4. Qualitative evaluation

For additional qualitative evaluation on MJHQ dataset[3], please see the `html` pages in the zip file and Figure 2.

#### A.5. Quantitative evaluation

Here we report additional evaluation of the proposed approach with a wider set of metrics. In particular, we rely on VQA [4] to measure the adherence of the generated samples to the input prompts. We measure the image quality with CLIP-IQA [8].

Table 4 shows FraQAT outperforms even the strongest QAT baseline we developed namely SVDQAT, with overall higher gains for SD3.5-Medium and PixArt-Σ for both test datasets.

#### A.6. Quantization schedule

To validate the benefits of a *Fractional* quantization schedule (Table 3) we compare it with its *Integer* counterpart ( $8 \rightarrow 7 \rightarrow 6 \rightarrow 5 \rightarrow 4$ ), and a simpler progressive schedule ( $16 \rightarrow 8 \rightarrow 4$ ). For a fair comparison, all experiments have the same computational budget. We measure the validation loss across training and plot it in Figure 3. The integer and simple schedules perform comparably to each other. On the other hand, the Fractional schedule consistently outperforms the two competitors during training, resulting in a sensibly lower validation loss.

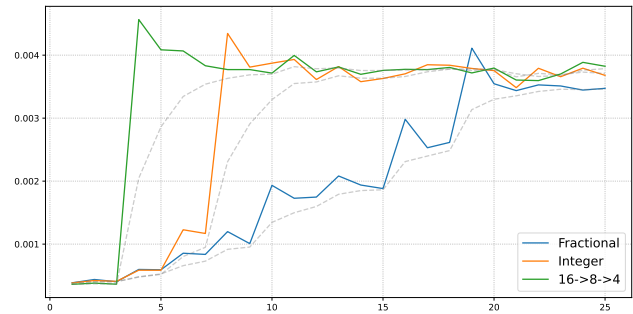


Figure 3. **Fractional schedule:** we train SD3.5-M using a simple progressive schedule (green), an integer schedule (orange), and a fractional schedule (blue). As seen in the graph, the Fractional schedule achieves a lower validation loss.

## B. Additional evaluation

### B.1. Language Model

The proposed method is agnostic to the architecture and the application. We apply FraQAT to Gemma2 2B IT [7]<sup>6</sup>. Start from the FP16 model – original –, then we quantize it to 4 bits in a similar fashion as we did with T2I models in the main paper. We follow the same schedule as in Section A.3. The quantized model (W4A8) is then compared with the original model.

As training set we rely on a subset of C4 dataset [5]: we pick randomly 384K samples for training and 38.4K samples for validation. The model is evaluated on two datasets in zero-shot fashion: BoolQ [2]<sup>7</sup>, and Commonsense QA [6]<sup>8</sup>. Table 5 shows minimal drop when FraQAT is applied to Gemma2 2B IT model. Therefore, proving FraQAT can be applied to Language Models as well as Vision Models.

<sup>3</sup><https://pytorch.org/>

<sup>4</sup><https://lightning.ai/docs/pytorch/stable/>

<sup>5</sup><https://github.com/pytorch/ao>

<sup>6</sup><https://huggingface.co/google/gemma-2-2b-it>

<sup>7</sup><https://huggingface.co/datasets/google/boolq>

<sup>8</sup>[https://huggingface.co/datasets/allenai/social\\_i\\_qa](https://huggingface.co/datasets/allenai/social_i_qa)

Table 2. **Hyper-parameters:** Detailed hyper-parameters required to replicate all experiments.

		SD3.5-M			Sana 600M			PixArt-Σ		
		lr	batch size	low rank	lr	batch size	low rank	lr	batch size	low rank
SVDQAT	W4A8	10 <sup>-5</sup>	128	32	10 <sup>-6</sup>	128	16	10 <sup>-6</sup>	256	16
vQAT	W4A8	10 <sup>-5</sup>	256	-	10 <sup>-6</sup>	128	-	10 <sup>-6</sup>	128	-
FraQAT	W4A8	10 <sup>-6</sup>	256	-	10 <sup>-7</sup>	128	-	10 <sup>-6</sup>	128	-

Table 3. **Precision schedule:** During training we progressively reduce the precision following the prescribed schedule.

eration. Advances in Neural Information Processing Systems, 36:15903–15935, 2023. 4

Precision	8	7	6	5.5	5	4.75	4.5	4.25	4
# epochs	1	1	1	1	1	2	2	2	14

References

[1] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation. In European Conference on Computer Vision, pages 74–91. Springer, 2024. 1, 4

[2] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. arXiv preprint arXiv:1905.10044, 2019. 2

[3] Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2. 5: Three insights towards enhancing aesthetic quality in text-to-image generation. arXiv preprint arXiv:2402.17245, 2024. 2, 4

[4] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. arXiv preprint arXiv:2404.01291, 2024. 2, 4

[5] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research, 21(140):1–67, 2020. 2

[6] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. arXiv preprint arXiv:1811.00937, 2018. 2

[7] Gemma Team. Gemma. 2024. 2, 4

[8] Jianyi Wang, Kelvin CK Chan, and Chen Change Loy. Exploring clip for assessing the look and feel of images. In Proceedings of the AAAI conference on artificial intelligence, pages 2555–2563, 2023. 2, 4

[9] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image gen-

Table 4. **Qualitative evaluation:** we evaluate FraQAT using a *fractional quantization schedule* on PixArt Evaluation dataset [1] and MidJourney HQ Evaluation dataset [3] measuring FID, and CLIP-FID wrt the original model, CLIP-IQA [8], ImageReward (IR) [9], and VQA [4].

PixArt-Σ																					
		SD3.5 Medium					Sana 600M					PixArt-Σ					Flux-schnell				
Method	Precision	FID ↓	CLIP FID ↓	CLIP IQA ↑	IR ↑	VQA score ↑	FID ↓	CLIP FID ↓	CLIP IQA ↑	IR ↑	VQA score ↑	FID ↓	CLIP FID ↓	CLIP IQA ↑	IR ↑	VQA score ↑	FID ↓	CLIP FID ↓	CLIP IQA ↑	IR ↑	VQA score ↑
Dynamic Q.	W4A8	9.36	2.08	0.44	0.56	0.84	2.22	0.24	<b>0.46</b>	0.57	<b>0.82</b>	13.35	6.19	0.44	0.35	0.82	8.17	1.13	<b>0.43</b>	-0.73	0.77
DiTAS	W4A8	27.93	13.77	<b>0.47</b>	0.41	0.82	12.87	4.58	0.45	<b>0.62</b>	<b>0.82</b>	7.30	3.95	<b>0.46</b>	<b>0.84</b>	<b>0.86</b>	-	-	-	-	-
SVDQuant	W4A16	14.42	3.14	0.42	0.66	0.85	2.43	0.24	0.43	0.60	<b>0.82</b>	6.80	2.02	0.43	0.79	<b>0.86</b>	<b>2.26</b>	0.36	0.42	0.84	0.85
SVDQAT	W4A8	2.57	0.28	0.45	0.80	0.85	<b>1.93</b>	<b>0.13</b>	0.43	0.48	<b>0.82</b>	5.38	1.48	0.43	0.76	<b>0.86</b>	-	-	-	-	-
vQAT	W4A8	2.67	0.31	0.44	0.78	0.85	2.13	0.16	0.43	0.45	0.81	7.00	2.52	0.45	0.79	0.85	3.40	0.66	0.41	<b>0.87</b>	<b>0.86</b>
<b>FraQAT</b>	<b>W4A8</b>	<b>2.54</b>	<b>0.27</b>	<b>0.45</b>	<b>0.82</b>	<b>0.86</b>	2.17	0.19	0.42	0.48	<b>0.82</b>	<b>4.48</b>	<b>1.07</b>	<b>0.45</b>	0.79	<b>0.86</b>	2.55	<b>0.30</b>	0.41	0.86	0.85

MJHQ																					
		SD3.5 Medium					Sana 600M					PixArt-Σ					Flux-schnell				
Method	Precision	FID ↓	CLIP FID ↓	CLIP IQA ↑	IR ↑	VQA score ↑	FID ↓	CLIP FID ↓	CLIP IQA ↑	IR ↑	VQA score ↑	FID ↓	CLIP FID ↓	CLIP IQA ↑	IR ↑	VQA score ↑	FID ↓	CLIP FID ↓	CLIP IQA ↑	IR ↑	VQA score ↑
Dynamic Q.	W4A8	10.29	2.11	0.44	0.65	0.79	2.40	0.28	<b>0.45</b>	0.63	0.74	15.04	5.55	0.43	0.44	0.74	8.66	1.24	0.42	-0.90	0.65
DiTAS	W4A8	32.04	14.06	<b>0.47</b>	0.41	0.73	12.91	5.59	<b>0.45</b>	<b>0.68</b>	<b>0.75</b>	8.63	4.07	<b>0.46</b>	<b>1.04</b>	0.80	-	-	-	-	-
SVDQuant	W4A16	15.10	3.06	0.42	0.78	0.78	2.48	0.25	0.42	0.62	<b>0.75</b>	6.95	1.71	0.43	0.99	0.80	<b>2.41</b>	0.41	<b>0.42</b>	0.96	0.79
SVDQAT	W4A8	2.85	<b>0.32</b>	0.45	0.91	0.80	<b>2.04</b>	<b>0.16</b>	0.43	0.53	0.74	5.83	1.44	0.43	0.96	<b>0.81</b>	-	-	-	-	-
vQAT	W4A8	3.01	0.37	0.44	0.89	0.80	2.13	0.20	0.43	0.47	0.74	7.38	2.12	0.44	0.99	0.80	3.56	0.73	0.41	<b>0.99</b>	<b>0.80</b>
<b>FraQAT</b>	<b>W4A8</b>	<b>2.78</b>	<b>0.32</b>	<b>0.45</b>	<b>0.96</b>	<b>0.81</b>	2.34	0.24	0.42	0.50	0.74	<b>4.95</b>	<b>1.05</b>	0.44	0.97	0.80	2.55	<b>0.39</b>	0.41	<b>0.99</b>	<b>0.80</b>

Table 5. **Evaluation on Language Models:** We apply FraQAT to Gemma2 2B IT [7] exactly as we did to T2I models. The quantized model is evaluated on Common Sense QA and BoolQ datasets. The resulting model has minimal drop from original language model.

Model	Precision	Common Sense QA ↑	BoolQ ↑	COQA ↑
Original	(W16A32)	0.70 ± 0.01	0.76 ± 0.01	0.66 ± 0.01
FraQAT	(W4A8)	0.69 ± 0.01	0.72 ± 0.01	0.70 ± 0.01