

1.58-bit FLUX

Chenglin Yang¹, Celong Liu¹, Xueqing Deng¹, Dongwon Kim²,
Xing Mei¹, Xiaohui Shen¹, Liang-Chieh Chen¹

¹ByteDance ²POSTECH

<https://chenglin-yang.github.io/1.58bit.flux.github.io/>



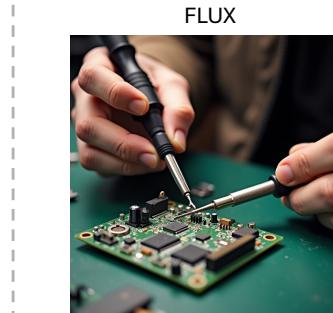
A cat made of sea water walking in a library.



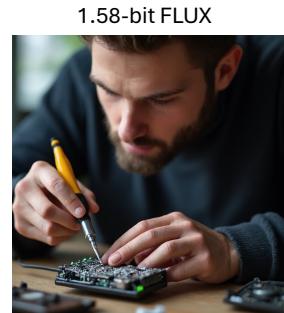
A dragon made of fire is spiraling in the sky above the city.



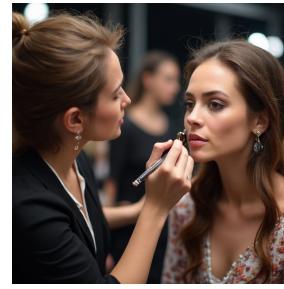
A bird behind a rabbit.



A man is holding a soldering iron and repairing a broken electronic device.



The makeup artist is creating a dramatic look for a fashion show.



A television on the top of a bird.

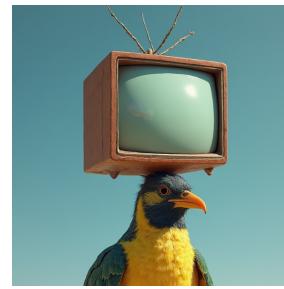


Figure 1. **Visual comparisons between FLUX and 1.58-bit FLUX.** 1.58-bit FLUX demonstrates comparable generation quality to FLUX while employing 1.58-bit quantization, where 99.5% of the 11.9B parameters in the vision transformer are constrained to the values +1, -1, or 0. For consistency, all images in each comparison are generated using the same latent noise input. 1.58-bit FLUX utilizes a custom 1.58-bit kernel. Additional visual comparisons are provided in Fig. 3 and Fig. 4.

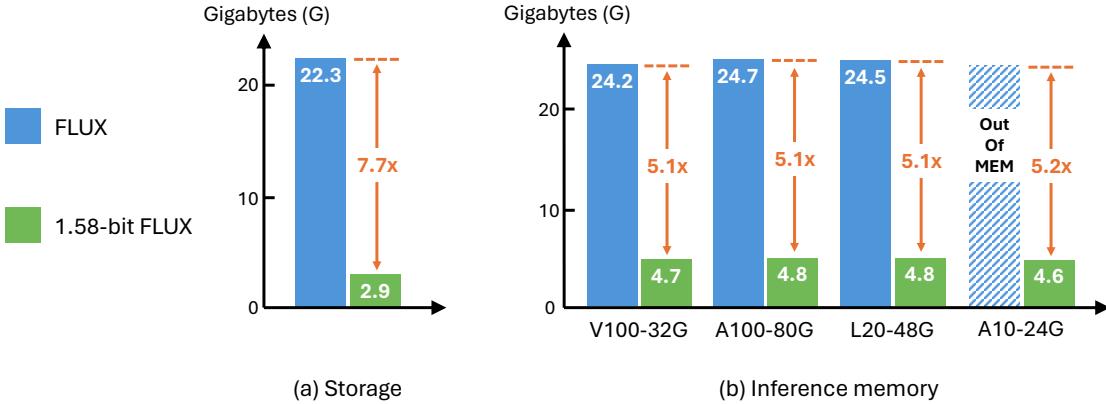


Figure 2. **Efficiency measurements on the vision transformer component of FLUX and 1.58-bit FLUX.** The measurements are based on generating a single image with 50 inference steps. (a) 1.58-bit FLUX reduces checkpoint storage by $7.7\times$ compared to FLUX. (b) 1.58-bit FLUX achieves a $5.1\times$ reduction in inference memory usage across various GPU types. The x-axis labels, $m-nG$, represent GPU type m with a maximum memory capacity of n Gigabytes (G).

Abstract

We present 1.58-bit FLUX, the first successful approach to quantizing the state-of-the-art text-to-image generation model, FLUX.1-dev, using 1.58-bit weights (i.e., values in $\{-1, 0, +1\}$) while maintaining comparable performance for generating 1024×1024 images. Notably, our quantization method operates without access to image data, relying solely on self-supervision from the FLUX.1-dev model. Additionally, we develop a custom kernel optimized for 1.58-bit operations, achieving a $7.7\times$ reduction in model storage, a $5.1\times$ reduction in inference memory, and improved inference latency. Extensive evaluations on the GenEval and T2I Compbench benchmarks demonstrate the effectiveness of 1.58-bit FLUX in maintaining generation quality while significantly enhancing computational efficiency.

1. Introduction

Recent text-to-image (T2I) models, including DALLE 3 [3], Adobe Firefly 3 [1], Stable Diffusion 3 [14], Midjourney v6.1 [41], Ideogram v2 [26], PlayGround V3 [36], FLUX 1.1 [29], Recraft V3 [2], have demonstrated remarkable generative capabilities, making them highly promising for real-world applications. However, their immense parameter counts, often in the billions, and high memory requirements during inference pose significant challenges for deployment on resource-constrained devices such as mobile platforms.

In this work, we address these challenges by exploring extreme low-bit quantization of T2I models. Among available state-of-the-art models, we select FLUX.1-dev [30] as our quantization target due to its public availability and

competitive performance.¹ Specifically, we quantize the weights of the vision transformer in FLUX to 1.58 bits without relying on mixed-precision schemes or access to image data. The quantization restricts linear layer weights to the values $\{+1, 0, -1\}$, akin to the BitNet b1.58 [40] approach. Unlike BitNet, which involves training large language models from scratch, our method operates as a post-training quantization solution for T2I models.

This quantization reduces model storage by $7.7\times$, as the 1.58-bit weights are stored using 2-bit signed integers, compressing them from 16-bit precision. To further enhance inference efficiency, we introduce a custom kernel optimized for low-bit computation. This kernel reduces inference memory usage by over $5.1\times$ and improves inference latency, as detailed in Fig. 2 and Tab. 3. Comprehensive evaluations on T2I benchmarks, including GenEval [17] and T2I Compbench [23], reveal that 1.58-bit FLUX achieves comparable performance to full-precision FLUX, as shown in Tab. 1 and Tab. 2.

Our contributions can be summarized as follows:

- We introduce 1.58-bit FLUX, the first quantized model to reduce 99.5% of FLUX vision transformer parameters (11.9B in total) to 1.58 bits without requiring image data, significantly lowering storage requirements.
- We develop an efficient linear kernel optimized for 1.58-bit computation, enabling substantial memory reduction and inference speedup.
- We demonstrate that 1.58-bit FLUX maintains performance comparable to the full-precision FLUX model on challenging T2I benchmarks.

This work represents a significant step forward in making high-quality T2I models practical for deployment on

¹We refer FLUX to FLUX.1-dev in this paper for simplicity.

Model	Avg.	Color	Shape	Texture	2D Spatial	3D Spatial	Numeracy	Non-spatial	Complex
		B-VQA	B-VQA	B-VQA	UniDet	UniDet	UniDet	S-CoT	S-CoT
Stable XL [45]	0.5255	0.5879	0.4687	0.5299	0.2133	0.3566	0.4988	0.7673	0.7817
Pixart- α -ft [6]	0.5586	0.6690	0.4927	0.6477	0.2064	0.3901	0.5058	0.7747	0.7823
FLUX	0.5876	0.7529	0.5056	0.6299	0.2791	0.4014	0.6131	0.7807	0.7380
1.58-bit FLUX (w/o kernel)	0.5806	0.7358	0.4900	0.6151	0.2781	0.4037	0.6089	0.7807	0.7327
1.58-bit FLUX	0.5812	0.7390	0.4910	0.6162	0.2757	0.4049	0.6137	0.7793	0.7300

Table 1. **Evaluations on T2I CompBench.** 1.58-bit FLUX (w/o kernel) indicates no efficient kernel is applied.

Model	Overall	Single object	Two object	Counting	Colors	Position	Color attribution
Stable XL [45]	0.55	0.98	0.74	0.39	0.85	0.15	0.23
PlayGroundv2.5 [31]	0.56	0.98	0.77	0.52	0.84	0.11	0.17
FLUX	0.66	0.98	0.81	0.74	0.79	0.22	0.45
1.58-bit FLUX (w/o kernel)	0.64	0.98	0.79	0.69	0.77	0.20	0.43
1.58-bit FLUX	0.64	0.98	0.77	0.68	0.79	0.21	0.44

Table 2. **Evaluations on GenEval.** 1.58-bit FLUX (w/o kernel) indicates no efficient kernel is applied.

GPU	FLUX	1.58-bit FLUX	Improvements
V100	74.8	73.6	1.6%
A100	26.4	25.0	5.3%
L20	90.2	78.3	13.2%
A10	OOM	84.4	—

Table 3. **Latency measurements on the vision transformer component of FLUX and 1.58-bit FLUX.** The measurements are obtained by generating one image with 50 inference steps. OOM means out of memory.

memory- and latency-constrained devices.

2. Related Work

Quantization is a widely adopted technique for reducing model size and enhancing inference efficiency, as demonstrated in numerous studies [4, 10, 13, 15, 18, 19, 25, 27, 42, 43, 53]. It has proven particularly effective for serving large language models (LLMs) [8, 12, 16, 28, 34, 35, 39, 47, 54, 56, 59, 63], enabling significant resource savings without compromising performance. Recent advancements include both post-training quantization (PTQ) methods [16, 56, 59], which adjust pre-trained models for efficient deployment, and Quantization Aware Training (QAT) approaches that fine-tune the model to low bits from pre-trained checkpoints [9, 21, 39, 44] or train the model from scratch [51]. For instance, BitNet b1.58 [40] employs weights in linear layers restricted to three values {+1, 0,

-1}, facilitating highly efficient inference, particularly on CPUs [52]. These developments highlight the potential of quantization to address the computational challenges associated with deploying large-scale models.

For image generation models, prior work has explored various quantization techniques for diffusion models [5, 7, 20, 22, 24, 33, 38, 46, 49, 50, 55, 57, 58], including Hadamard transformations [37], vector quantization [11], floating-point quantization [37], mixed bit-width allocation [62], diverse quantization metrics [61], multiple-stage fine-tuning [48] and low-rank branch [32]. These approaches aim to optimize model efficiency while maintaining performance. In this work, we focus on post-training 1.58-bit quantization of FLUX, a state-of-the-art open-source text-to-image (T2I) model. Notably, our method achieves efficient quantization without relying on any tuning image data and is complemented by optimized inference techniques.

3. Experimental Results

3.1. Settings

Quantization. We utilize a calibration dataset comprising prompts from the Parti-1k dataset [60] and the training split of T2I CompBench [23], totaling 7,232 prompts. This process is entirely image-data-free, requiring no additional datasets. The quantization reduces the weights of all linear layers in the FluxTransformerBlock and FluxSingleTransformerBlock of FLUX to 1.58 bits, covering 99.5% of the

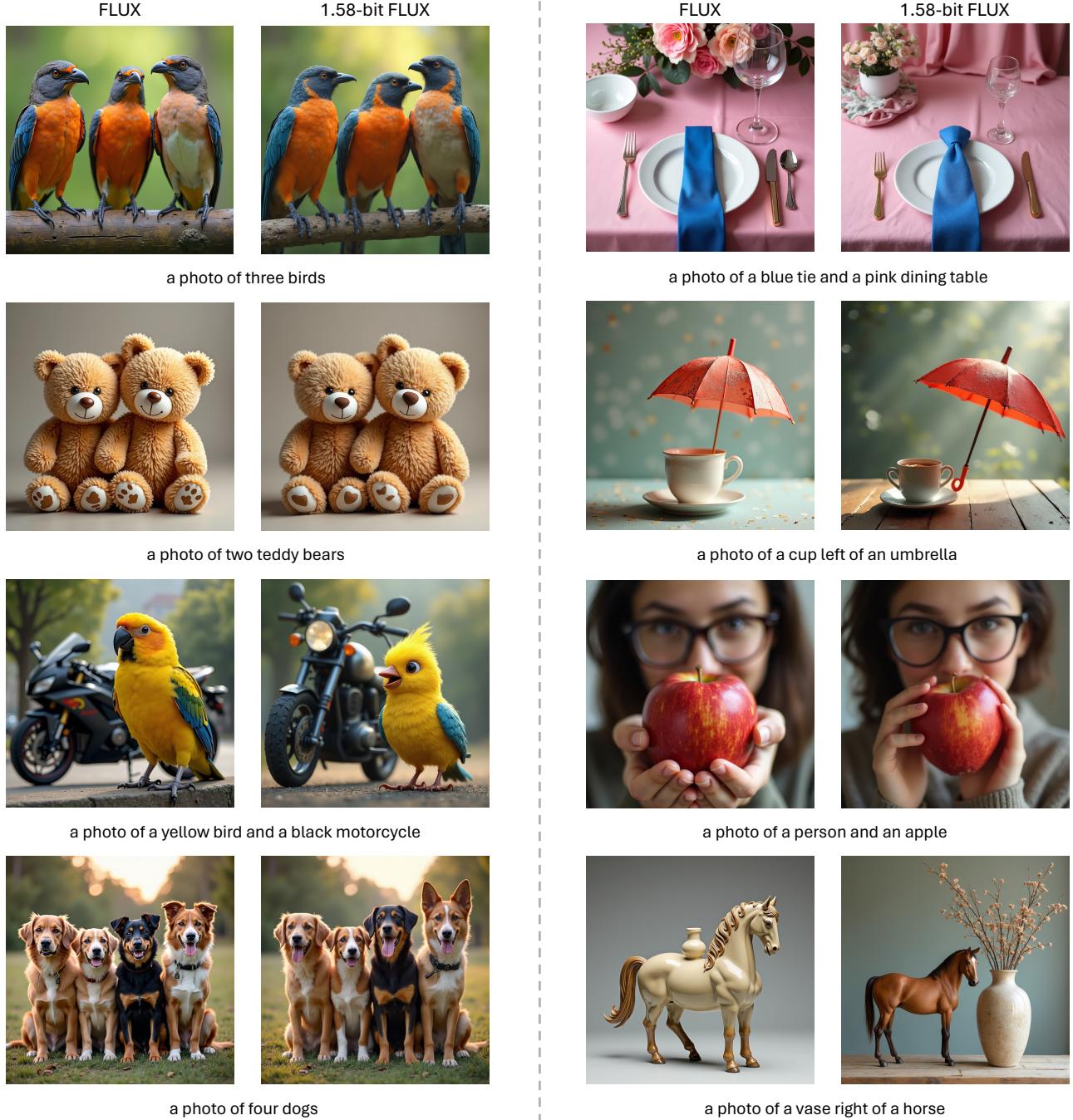


Figure 3. Visual comparisons between FLUX and 1.58-bit FLUX on GenEval dataset. 1.58-bit FLUX demonstrates comparable generation quality to FLUX while employing 1.58-bit quantization, where 99.5% of the 11.9B parameters in the vision transformer are constrained to the values +1, -1, or 0. For consistency, all images in each comparison are generated using the same latent noise input. 1.58-bit FLUX utilizes a custom 1.58-bit kernel.

model’s total parameters.

Evaluation. We evaluate both FLUX and 1.58-bit FLUX on the GenEval dataset [17] and the validation split of T2I

CompBench [23], following the official image generation pipeline. The GenEval dataset consists of 553 prompts, with four images generated per prompt. The T2I Comp-

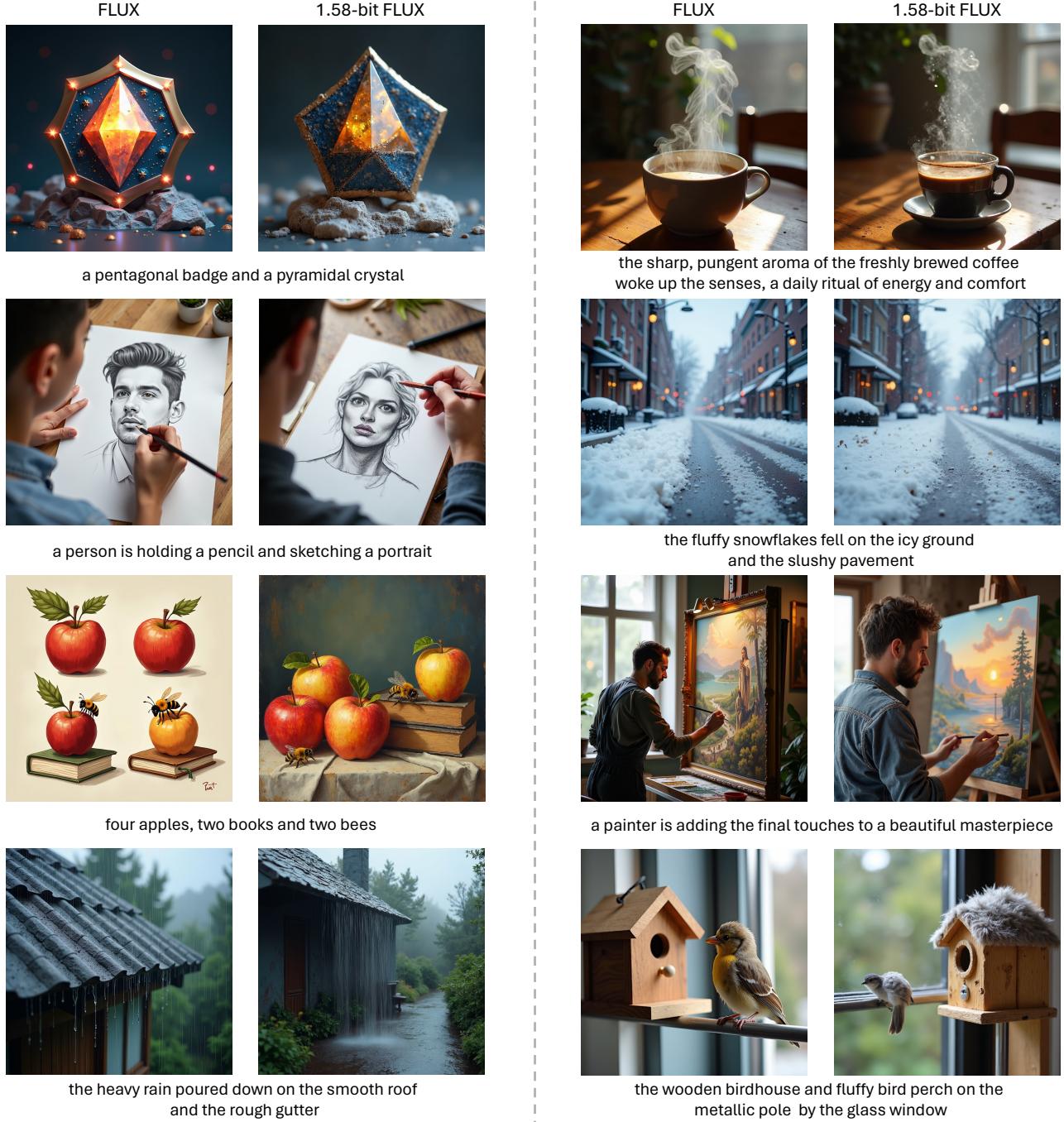


Figure 4. **Visual comparisons between FLUX and 1.58-bit FLUX on the validation split of T2I CompBench.** 1.58-bit FLUX demonstrates comparable generation quality to FLUX while employing 1.58-bit quantization, where 99.5% of the 11.9B parameters in the vision transformer are constrained to the values +1, -1, or 0. For consistency, all images in each comparison are generated using the same latent noise input. 1.58-bit FLUX utilizes a custom 1.58-bit kernel.

Bench validation split includes eight categories, each containing 300 prompts, with 10 images generated per prompt, yielding a total of 24,000 images for evaluation. All images

are generated at a resolution of 1024×1024 for both FLUX and 1.58-bit FLUX.

3.2. Results

Performance. Comparable performances between 1.58-bit FLUX and FLUX on T2I Compbench and GenEval are observed in Tab. 1 and Tab. 2, respectively. The minor differences observed before and after applying our linear kernel further demonstrate the accuracy of our implementation.

Efficiency. Significant efficiency gains are observed in both model storage and inference memory, as shown in Fig. 2. For inference latency, as illustrated in Tab. 3, even greater improvements are achieved when running 1.58-bit FLUX on lower-performing yet deployment-friendly GPUs, such as the L20 and A10.

4. Conclusion and Discussion

This work introduced 1.58-bit FLUX, in which 99.5% of the transformer parameters are quantized to 1.58 bits. With our custom computation kernels, 1.58-bit FLUX achieves a $7.7 \times$ reduction in model storage and more than a $5.1 \times$ reduction in inference memory usage. Despite these compression gains, 1.58-bit FLUX demonstrates comparable performance on T2I benchmarks and maintains high visual quality. We hope that 1.58-bit FLUX inspires the community to develop more robust models for mobile devices.

We discuss the current limitations of 1.58-bit FLUX below, which we plan to address in future work:

- **Limitations on speed improvements.** Although 1.58-bit FLUX reduces model size and memory consumption, its latency is not significantly improved due to the absence of activation quantization and lack of further optimized kernel implementations. Given our promising results, we hope to inspire the community to develop custom kernel implementation for 1.58-bit models.
- **Limitations on visual qualities.** Fig. 1, Fig. 3 and Fig. 4 show that 1.58-bit FLUX can generate vivid and realistic images closely aligned with the given text prompts. However, it still lags behind the original FLUX model in rendering fine details at very high resolutions. We aim to address this gap in future research.

References

- [1] Adobe. <https://news.adobe.com/news/news-details/2024/adobe-introduces-firefly-image-3-foundation-model-to-take-creative-exploration-and-ideation-to-new-heights.2024>. 2
- [2] Recraft AI. <https://www.recraft.ai/blog/recraft-introduces-a-revolutionary-ai-model-that-thinks-in-design-language.2024>. 2
- [3] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf>, 2(3):8, 2023. 2
- [4] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13169–13178, 2020. 3
- [5] Cheng Chen, Christina Giannoula, and Andreas Moshovos. Low-bitwidth floating point quantization for efficient high-quality diffusion models. *arXiv preprint arXiv:2408.06995*, 2024. 3
- [6] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023. 3
- [7] Lei Chen, Yuan Meng, Chen Tang, Xinzhu Ma, Jingyan Jiang, Xin Wang, Zhi Wang, and Wenwu Zhu. Q-dit: Accurate post-training quantization for diffusion transformers. *arXiv preprint arXiv:2406.17343*, 2024. 3
- [8] Mengzhao Chen, Yi Liu, Jiahao Wang, Yi Bin, Wenqi Shao, and Ping Luo. Prefixquant: Static quantization beats dynamic through prefixed outliers in llms. *arXiv preprint arXiv:2410.05265*, 2024. 3
- [9] Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. Efficientqat: Efficient quantization-aware training for large language models. *arXiv preprint arXiv:2407.11062*, 2024. 3
- [10] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. Low-bit quantization of neural networks for efficient inference. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3009–3018. IEEE, 2019. 3
- [11] Juncan Deng, Shuaiting Li, Zeyu Wang, Hong Gu, Kedong Xu, and Kejie Huang. Vq4dit: Efficient post-training vector quantization for diffusion transformers. *arXiv preprint arXiv:2408.17131*, 2024. 3
- [12] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35:30318–30332, 2022. 3
- [13] Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Hawq: Hessian aware quantization of neural networks with mixed-precision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 293–302, 2019. 3
- [14] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024. 2
- [15] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019. 3
- [16] Elias Frantar, Saleh Ashkboos, Torsten Hoefer, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022. 3

- [17] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 4
- [18] Ruihao Gong, Xianglong Liu, Shenghu Jiang, Tianxiang Li, Peng Hu, Jiazen Lin, Fengwei Yu, and Junjie Yan. Differentiable soft quantization: Bridging full-precision and low-bit neural networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4852–4861, 2019. 3
- [19] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. 3
- [20] Yefei He, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Efficientdm: Efficient quantization-aware fine-tuning of low-bit diffusion models. *arXiv preprint arXiv:2310.03270*, 2023. 3
- [21] Yefei He, Zhenyu Lou, Luoming Zhang, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Bivit: Extremely compressed binary vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5651–5663, 2023. 3
- [22] Yefei He, Luping Liu, Jing Liu, Weijia Wu, Hong Zhou, and Bohan Zhuang. Ptqd: Accurate post-training quantization for diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 3
- [23] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-combench: A comprehensive benchmark for open-world compositional text-to-image generation. *Advances in Neural Information Processing Systems*, 36:78723–78747, 2023. 2, 3, 4
- [24] Yushi Huang, Ruihao Gong, Jing Liu, Tianlong Chen, and Xianglong Liu. Tfmq-dm: Temporal feature maintenance quantization for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7362–7371, 2024. 3
- [25] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks. *Advances in Neural Information Processing Systems*, 29, 2016. 3
- [26] Ideogram. <https://updates.midjourney.com/version-6-1/>. 2024. 2
- [27] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713, 2018. 3
- [28] Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W Mahoney, and Kurt Keutzer. Squeezellm: Dense-and-sparse quantization. *arXiv preprint arXiv:2306.07629*, 2023. 3
- [29] Black Forest Labs. <https://blackforestlabs.ai/announcements/>. 2024. 2
- [30] Black Forest Labs. <https://github.com/black-forest-labs/flux>. 2024. 2
- [31] 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. 3
- [32] 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. 3
- [33] Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17535–17545, 2023. 3
- [34] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6:87–100, 2024. 3
- [35] Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. Qserve: W4a8kv4 quantization and system co-design for efficient llm serving. *arXiv preprint arXiv:2405.04532*, 2024. 3
- [36] Bingchen Liu, Ehsan Akhgari, Alexander Visheratin, Aleks Kamko, Linmiao Xu, Shivam Shriraos, Chase Lambert, Joao Souza, Suhail Doshi, and Daiqing Li. Playground v3: Improving text-to-image alignment with deep-fusion large language models. *arXiv preprint arXiv:2409.10695*, 2024. 2
- [37] Wenxuan Liu and Sai Qian Zhang. Hq-dit: Efficient diffusion transformer with fp4 hybrid quantization. *arXiv preprint arXiv:2405.19751*, 2024. 3
- [38] Xuewen Liu, Zhikai Li, Junrui Xiao, and Qingyi Gu. Enhanced distribution alignment for post-training quantization of diffusion models. *arXiv preprint arXiv:2401.04585*, 2024. 3
- [39] Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. Llm-qat: Data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*, 2023. 3
- [40] Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wen-hui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 2024. 2, 3
- [41] Midjourney. <https://updates.midjourney.com/version-6-1/>. 2024. 2
- [42] 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. 3
- [43] 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. 3

- [44] Markus Nagel, Marios Fournarakis, Yelysei Bondarenko, and Tijmen Blankevoort. Overcoming oscillations in quantization-aware training. In *International Conference on Machine Learning*, pages 16318–16330. PMLR, 2022. 3
- [45] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 3
- [46] 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. 3
- [47] Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omnipquant: Omnidirectionally calibrated quantization for large language models. *arXiv preprint arXiv:2308.13137*, 2023. 3
- [48] Yang Sui, Yanyu Li, Anil Kag, Yerlan Idelbayev, Junli Cao, Ju Hu, Dhritiman Sagar, Bo Yuan, Sergey Tulyakov, and Jian Ren. Bitsfusion: 1.99 bits weight quantization of diffusion model. *arXiv preprint arXiv:2406.04333*, 2024. 3
- [49] Siao Tang, Xin Wang, Hong Chen, Chaoyu Guan, Zewen Wu, Yansong Tang, and Wenwu Zhu. Post-training quantization with progressive calibration and activation relaxing for text-to-image diffusion models. In *European Conference on Computer Vision*, pages 404–420. Springer, 2025. 3
- [50] Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. Towards accurate post-training quantization for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16026–16035, 2024. 3
- [51] H Wang, S Ma, L Dong, S Huang, H Wang, L Ma, F Yang, R Wang, Y Wu, and F Wei. Bitnet: Scaling 1-bit transformers for large language models. arxiv. *arXiv preprint arXiv:2310.11453*, 2023. 3
- [52] Jinheng Wang, Hansong Zhou, Ting Song, Shaoguang Mao, Shuming Ma, Hongyu Wang, Yan Xia, and Furu Wei. 1-bit ai infra: Part 1.1, fast and lossless bitnet b1. 58 inference on cpus. *arXiv preprint arXiv:2410.16144*, 2024. 3
- [53] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8612–8620, 2019. 3
- [54] Xiuying Wei, Yunchen Zhang, Xiangguo Zhang, Ruihao Gong, Shanghang Zhang, Qi Zhang, Fengwei Yu, and Xianlong Liu. Outlier suppression: Pushing the limit of low-bit transformer language models. *Advances in Neural Information Processing Systems*, 35:17402–17414, 2022. 3
- [55] Junyi Wu, Haoxuan Wang, Yuzhang Shang, Mubarak Shah, and Yan Yan. Ptq4dit: Post-training quantization for diffusion transformers. *arXiv preprint arXiv:2405.16005*, 2024. 3
- [56] 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 *International Conference on Machine Learning*, pages 38087–38099. PMLR, 2023. 3
- [57] Yuewei Yang, Xiaoliang Dai, Jialiang Wang, Peizhao Zhang, and Hongbo Zhang. Efficient quantization strategies for latent diffusion models. *arXiv preprint arXiv:2312.05431*, 2023. 3
- [58] Yuzhe Yao, Feng Tian, Jun Chen, Haonan Lin, Guang Dai, Yong Liu, and Jingdong Wang. Timestep-aware correction for quantized diffusion models. *arXiv preprint arXiv:2407.03917*, 2024. 3
- [59] Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in Neural Information Processing Systems*, 35:27168–27183, 2022. 3
- [60] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunnar Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2(3):5, 2022. 3
- [61] Tianchen Zhao, Tongcheng Fang, Enshu Liu, Wan Rui, Widyadewi Soedarmadji, Shiyao Li, Zinan Lin, Guohao Dai, Shengen Yan, Huazhong Yang, et al. Vudit-q: Efficient and accurate quantization of diffusion transformers for image and video generation. *arXiv preprint arXiv:2406.02540*, 2024. 3
- [62] Tianchen Zhao, Xuefei Ning, Tongcheng Fang, Enshu Liu, Guyue Huang, Zinan Lin, Shengen Yan, Guohao Dai, and Yu Wang. Mixdq: Memory-efficient few-step text-to-image diffusion models with metric-decoupled mixed precision quantization. *arXiv preprint arXiv:2405.17873*, 2024. 3
- [63] Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, and Baris Kasikci. Atom: Low-bit quantization for efficient and accurate llm serving. *Proceedings of Machine Learning and Systems*, 6:196–209, 2024. 3