SD_Quant: Lessons in Post-Training Quantization for Diffusion Models

Elvin Lo
Harvard College
elo@college.harvard.edu

Jaray Liu
Harvard College
jarayliu@college.harvard.edu

Carlos Luz
Harvard College
cluz@college.harvard.edu

Anmay Gupta
Harvard College
anmaygupta@college.harvard.edu

***Abstract*—Diffusion models have shown exceptional image generation capabilities, but are computationally expensive. In this work, we explore post-training quantization (PTQ) techniques to compress and accelerate diffusion models. We improve on previous work by implementing rotation-based outlier reduction with random Hadamard transforms, dynamic activation quantization, and beginning to explore clustering-based weight quantization. Our work is the first to analyze which outlier reduction methods are appropriate for diffusion models, determining rotations to be suitable. We validate the efficacy of rotation-based outlier reduction in U-Net-based diffusion models, almost completely preserving full-precision performance of Stable Diffusion 2 in W8A8, achieving a 32.90 CLIP score compared to the full-precision 33.04. Furthermore, our preliminary work with clustering-based weight quantization is the most successful attempt yet at pushing diffusion model quantization into the W4A4 regime, achieving an impressive CLIP score of 29.30.**

***Index Terms*—quantization, diffusion models, outlier reduction**

I. Introduction

Diffusion models lead in image generation tasks, iteratively denoising Gaussian noise into high-fidelity image samples. However, their high computational cost limits their real-world deployment on edge devices. Post-training quantization (PTQ) decreases the bit-precision of weights and activations, compressing model sizes and inference costs. PTQ methods have had wide success in deep neural networks, with especially significant efforts dedicated to PTQ of LLMs. However, effectively quantizing diffusion models poses unique challenges, as these models involve iterative sampling steps which cause quantization error to propagate and increase the difficulty of quantizing activations since the activation distributions vary across timesteps [2].

In this work, we take steps toward determining an optimal PTQ strategy for compressing and accelerating diffusion models while retaining model accuracy. We propose SD_Quant, an approach to integer quantization of both weights and activations for diffusion models, incorporating several insights as illustrated in Fig. 1:

1) **Rotation-based outlier reduction:** We leverage random Hadamard transformations (RHTs) to minimize activation and weight outliers, and analyze why rotations are a suitable approach for diffusion models.

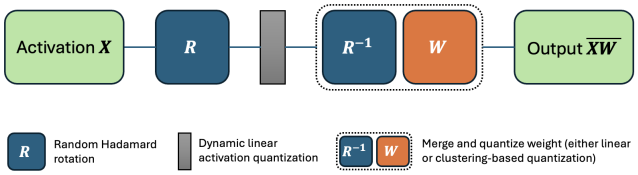

Fig. 1. Schematic of our quantization workflow in a linear layer. Given an input activation $\mathbf{X}$, we apply a random Hadamard transformation online, quantize the activation, and multiply by a quantized weight, in which the inverse rotation is already merged offline to maintain mathematical equivalence.

2) **Dynamic activation quantization:** We suggest determining activation quantization parameters dynamically, as we may collect the minimum and maximum values simultaneously while performing the rotations online.
3) **Clustering-based weight quantization:** We also conduct preliminary explorations on the possibility of quantizing each layer's weights by clustering them with K-means. This would turn the quantized multiplications into lookups in an offline-computed codebook.

By analyzing for the first time what outlier reduction methods are most suitable for diffusion models, our work paves the way for future quantization work on diffusion models. Our work is also the first to empirically validate the efficacy of rotation-based outlier reduction for preserving accuracy in U-Net-based diffusion models; indeed, in W8A8, we almost entirely preserve full-precision performance (32.90 CLIP score, versus 33.04) by implementing RHTs while doing dynamic activation quantization and symmetric uniform weight quantization. Furthermore, despite being preliminary, our work with cluster-based weight quantization combined with RHTs is the first to successfully push diffusion model quantization into the W4A4 regime, achieving an impressive CLIP score of 29.30 in W4A4.

II. Approach

Our quantization scheme has two components: outlier reduction techniques and quantization step choices. The former ensures the weight and activation distributions can be quantized without significant errors, while the latter makes our quantization approach robust to the temporally-evolving activation distribution.

A. Outlier reduction via rotation matrices

While outlier reduction methods typically require online computations that introduce inference overhead, experiments with methods like Q-Diffusion [2] and TDQ [9] have shown that pushing large diffusion models like Stable Diffusion 2 to lower bit-width regimes is not very feasible without outlier reduction. Recent works [5], [6] have naively applied popular outlier reduction techniques used in LLMs for the problem setting of diffusion transformers; however, they do not provide analysis into why such techniques may or may not work well in diffusion models, whose iterative sampling poses unique challenges. In this section, we discuss why rotation methods are an ideal choice for outlier reduction in diffusion models.

For some preliminaries: we focus on quantizing linear layers, as our work studying activation distribution statistics shows that linear layers have significantly higher variance compared to convolutional layers. When quantizing a linear layer with weights $\boldsymbol{W}$ and input activations $\boldsymbol{X}$, we normally compute the product $\overline{\boldsymbol{X}}\overline{\boldsymbol{W}}$ of their quantized counterparts. But to ease the quantization difficulty, we can reduce outliers first by instead computing

$$\overline{(\boldsymbol{X} \cdot \boldsymbol{T})} \cdot \overline{\left(\boldsymbol{T}^{-1} \cdot \boldsymbol{W}\right)}$$

where $\boldsymbol{T}$ is some invertible transformation. Attention layers may be quantized in the same way as linear layers by rotating the appropriate projection matrices. The application of $\boldsymbol{T}^{-1}$ to the weights can be done offline, since we only need to access the transformed weights at inference time; however, $\boldsymbol{T}$ must be applied to the activations at inference time. Hence, we need to choose some transformation $\boldsymbol{T}$ that reduces outliers but is also hardware-efficient to implement. Two possible choices are

- **Channel-wise smoothing** [7]: choose $\boldsymbol{T}$ to be a diagonal matrix, $\boldsymbol{T} = \mathrm{diag}(\boldsymbol{s})$; intuitively, this trades off quantization difficulty between activations and weights by trying to smooth out outlier feature channels.
- **Rotations** [3], [8]: choose $\boldsymbol{T}$ to be some rotation, often a random Hadamard transformation (RHT) since they can be implemented efficiently via butterfly algorithms; intuitively, rotations are a form of incoherence processing that reduce outliers by redistributing large values across the rows and columns of the matrices.

A previous work DiTAS [6] applies channel-wise smoothing to eliminate activation outliers in diffusion transformers. However, they neglect to evaluate whether channel-wise smoothing is a sensible approach for diffusion models. We revisit this question and deduce that channel-wise smoothing is *not* sensible for diffusion models. Channel-wise smoothing works well when the per-layer variance (across features channels) is large, but the per-channel variance (within a given channel, across the data distribution) is small, i.e., when outlier channels are consistently large [7]. However, we remark that this is *not* the case for diffusion models due to their nature of iterative sampling. This is because the activation magnitudes of a given channel in a diffusion model vary not just across the data, but also *across timesteps*, yielding high per-channel variance.

Meanwhile, random rotations increase the incoherence of the vectors to which they are applied (both weights and activations) to yield better tail bounds [8]. And importantly, we observe that the functionality of rotations does not rely on the assumption of low channel-wise variance in the activations; thus, rotations are a suitable method for quantizing diffusion models! In practice, Hadamard transformations can be implemented efficiently using a butterfly approach called the Fast Walsh–Hadamard transform; for a vector of length n, the rotation takes time only $O(n \log n)$.

B. Quantization choices: dynamic and clustering-based

After performing outlier reduction, it remains to determine the quantization step. Importantly, for diffusion models we need to quantize in a way that is robust across denoising timesteps, which is challenging because the activation distribution varies over time. While early approaches [2], [9] focus on linear quantization with static and timestep-static parameters respectively, we suggest a simpler approach: applying dynamic linear quantization, using the min and max values of activation tensors to determine the quantization parameters.

With dynamic quantization, adapting to changes in activation distribution across denoising iterations is not a concern, and the problem simplifies to quantizing the shape of the activation distribution. Our outlier reduction method addresses this by applying rotations online, so the minimum and maximum values may be determined simultaneously during or immediately after the final pass of the Fast Walsh-Hadamard transform with negligible added data movement and latency.

While we always quantize the activations linearly, we also conduct preliminary explorations on clustering-based weight quantization. That is, we use K-means with Lloyd's algorithm offline to cluster the weights in each layer into 2^{b_w} centroids, where b_w is the weight bit-width. In this case, multiplications between quantized weight and activation values correspond to lookups in a codebook with $2^{b_w+b_a}$ entries, telling us how to multiply each weight value by each activation value. At very low bit-widths, lookup-based quantization approaches have been computationally efficient, though it is uncertain whether our preliminary work will need further optimizations.

III. Implementation

We evaluated our quantization schemes on Stable Diffusion 2, a large diffusion model with 860 million parameters in its U-Net backbone. Quantization was emulated in PyTorch to assess accuracy degradation, measured by computing the CLIP score on the COCO 2017 validation dataset. The CLIP score represents the average cosine similarity between model-generated images and their corresponding text prompts. Due to computational constraints, we computed CLIP scores with 500 samples each and did not compute the Fréchet Inception Distance (FID) scores, another common metric for diffusion models. FID evaluations are numerically unstable and unreliable with sample sizes below 5000, as FID score is calculated via an unbiased estimator (and not by averaging a metric over model samples, like CLIP score).

TABLE I
CLIP SCORES FOR OUR QUANTIZATION SCHEMES

Quantization scheme	**Bit-width**	**CLIP Score**
Full precision (no quantization)	W32A32	33.04
Naive static activation quantization, symmetric uniform weight quantization	W8A8	5.31
Static activation quantization with uniform calibration, symmetric uniform weight quantization [2]	W8A8	30.46
	W4A8	1.95
	W4A4	0.0
Dynamic activation quantization, symmetric uniform weight quantization	W8A8	31.43
	W4A8	4.85
	W4A4	0.0
Dynamic activation quantization, symmetric uniform weight quantization, with RHT (us)	W8A8	32.90
	W4A8	12.20
	W4A4	5.40
Dynamic activation quantization, K-means weight quantization, with RHT (us)	W4A8	29.80
	W4A4	29.30

We remark that emulating quantization does not let us directly measure inference speedups, but prior work [5] also applying rotations demonstrated substantial improvements in computational efficiency.

IV. RESULTS

We benchmark our quantization schemes in bit-widths W8A8, W4A8, and W4A4, where WxAy denotes x-bit weights and y-bit activations. By employing RHTs, dynamic activation quantization, and a clustering-based weight quantization, we obtain a CLIP score of 29.30 at W4A4, a remarkable level of compression when comparing to our full precision baseline with score 33.04. With RHTs, dynamic activation quantization, and usual symmetric uniform weight quantization, we obtain a score of 32.90, almost completely preserving the full precision CLIP score, but we are not quite able to compress to W4A4 as doing so degrades the CLIP score to 12.90.

In contrast, previous work Q-Diffusion [2], which performs static activation quantization with carefully calibrated parameters, retains a reasonable score of 30.46 at W8A8, but cannot compress the weights further, yielding CLIP score 1.95 at W4A8. Using dynamic activation quantization improves these scores slightly, but cannot achieve W4A8 compression, demonstrating the importance of incoherence preprocessing.

Our method is the first to obtain such impressive results at W4A4 bit-width; the closest W4A4 result we have seen is SVDQuant [4] released just in November 2024, which experiments with the Stable Diffusion XL model and obtains CLIP score 26.5 with our same COCO evaluation dataset. While we have not had the time to implement their method on Stable Diffusion 2 for comparison or compare benchmarks on inference speed, we remark that our CLIP score is higher despite Stable Diffusion 2 having one third the parameter count of Stable Diffusion XL.

V. RELATED WORK

In this section, we overview previous works on PTQ for diffusion models. Initial works [1], [2] were amongst the first to observe the challenges in quantizing diffusion models posed by iterative sampling. To address these challenges, they determine new static quantization parameters per layer through careful calibration processes, e.g., uniformly sampling calibration data across diffusion steps or determining new sets of static parameters for each timestep [2], [9]. However, trying to quantize larger models to lower bit-widths has revealed that more advanced techniques are necessary to retain good performance.

Towards this end, recent related works on diffusion transformers apply some outlier reduction techniques that have been prominent in LLMs. As we previously discuss, DiTAS [6] applies channel-wise smoothing or SmoothQuant [7], while HQ-DiT [5] takes a similar approach to us and applies random rotations. However, we note that unlike our work, those works (1) do not analyze why or why not their methods are appropriate for diffusion models, (2) study different quantization settings (e.g., HQ-DiT focuses on floating-point quantization instead of the more commonly supported integer quantization), and (3) do not as extensively study quantization step choices, such as the K-means-based weight quantization we implement to obtain successful W4A4 compression for the first time.

VI. CONCLUSION

In this work, we explore PTQ for diffusion models, improving on previous work by introducing rotation-based outlier reduction with random Hadamard transforms, dynamic activation quantization, and beginning to explore clustering-based weight quantization. Though previous works apply outlier reduction techniques including rotations in diffusion transformers, our work is the first to analyze why rotations are the most suitable method for diffusion models, thereby paving the way for future quantization work on diffusion models which may further explore this direction, e.g., by learning optimal offline rotations for attention layers as in SpinQuant [3] in order to further improve outlier reduction without additional overhead. We are also the first to empirically validate the efficacy of rotation-based outlier reduction for preserving accuracy in U-Net-based diffusion models. Furthermore, our preliminary work combining clustering-based weight quantization with RHTs is the most successful attempt yet at pushing diffusion model quantization into the W4A4 regime, achieving an impressive CLIP score of 29.30.

Statement of Group Contributions

Elvin devised the initial project direction with random rotations, set up the codebase including benchmarking code and activation plotting, ran experiments, and contributed to the writeup. Jaray implemented the vast bulk of quantization techniques, first explored lookup-based quantization which yielded impressive results, and ran many, many experiments. Carlos visualized many results, especially studying the statistics of activations across layers, and contributed to the writeup. Anmay implemented static quantization and distributed multi-GPU experiments, contributed to the writeup, and helped the rest of the team to get set up on the FASRC cluster in the early stages of the project.

References

[1] Y. Shang, Z. Yuan, B. Xie, B. Wu, and Y. Yan, "Post-Training Quantization on Diffusion Models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 12345–12354.

[2] X. Li, Y. Wang, Y. Zhang, Y. Fu, Y. Wang, L. Zhang, and Y. Zhang, "Q-Diffusion: Quantizing Diffusion Models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 6789–6798.

[3] Z. Liu, C. Zhao, I. Fedorov, B. Soran, D. Choudhary, R. Krishnamoorthi, V. Chandra, Y. Tian, and T. Blankevoort, "SpinQuant: LLM Quantization with Learned Rotations," *arXiv preprint arXiv:2405.16406*, 2024. [Online]. Available: https://arxiv.org/abs/2405.16406.

[4] M. Li, Y. Lin, Z. Zhang, T. Cai, X. Li, J. Guo, E. Xie, C. Meng, J.-Y. Zhu, and S. Han, "SVDQuant: Absorbing Outliers by Low-Rank Components for 4-Bit Diffusion Models," *arXiv preprint arXiv:2411.05007*, 2024. [Online]. Available: https://arxiv.org/abs/2411.05007.

[5] W. Liu and S. Q. Zhang, "HQ-DiT: Efficient Diffusion Transformer with FP4 Hybrid Quantization," *arXiv preprint arXiv:2405.19751*, 2024. [Online]. Available: https://arxiv.org/abs/2405.19751.

[6] Z. Dong and S. Q. Zhang, "DiTAS: Quantizing Diffusion Transformers via Enhanced Activation Smoothing," *arXiv preprint arXiv:2409.07756*, 2024. [Online]. Available: https://arxiv.org/abs/2409.07756.

[7] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, "SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models," *arXiv preprint arXiv:2211.10438*, 2023. [Online]. Available: https://arxiv.org/abs/2211.10438.

[8] A. Tseng, J. Chee, Q. Sun, V. Kuleshov, and C. De Sa, "QuIP#: Even Better LLM Quantization with Hadamard Incoherence and Lattice Codebooks," *arXiv preprint arXiv:2402.04396*, 2024. [Online]. Available: https://arxiv.org/abs/2402.04396.

[9] J. So, J. Lee, D. Ahn, H. Kim, and E. Park, "Temporal Dynamic Quantization for Diffusion Models," *arXiv preprint arXiv:2306.02316*, 2023. [Online]. Available: https://arxiv.org/abs/2306.02316.