Post-Training Quantization for Audio Diffusion Transformers

Tanmay Khandelwal[1,2]*, Magdalena Fuentes*[2,3]

[1]Courant Institute of Mathematical Sciences, New York University, NY, USA
[2]MARL, New York University, NY, USA
[3]IDM, New York University, NY, USA

Abstract—**Diffusion Transformers (DiTs) enable high-quality audio synthesis but are often computationally intensive and require substantial storage, which limits their practical deployment. In this paper, we present a comprehensive evaluation of post-training quantization (PTQ) techniques for audio DiTs, analyzing the trade-offs between static and dynamic quantization schemes. We explore two practical extensions (1) a denoising-timestep-aware smoothing method that adapts quantization scales per-input-channel and timestep to mitigate activation outliers, and (2) a lightweight low-rank adapter (LoRA)-based branch derived from singular value decomposition (SVD) to compensate for residual weight errors. Using Stable Audio Open we benchmark W8A8 and W4A8 configurations across objective metrics and human perceptual ratings. Our results show that dynamic quantization preserves fidelity even at lower precision, while static methods remain competitive with lower latency. Overall, our findings show that low-precision DiTs can retain high-fidelity generation while reducing memory usage by up to 79%.**

1. INTRODUCTION

Diffusion models are a powerful type of generative model that excel at creating high-quality outputs in areas like audio generation [1], [2]. They are increasingly being adopted in music production [3]–[5] and sound design [6]. Compared to generative adversarial networks (GANs) and variational autoencoders (VAEs), diffusion models have more stable training and avoid issues like model collapse, making them a great choice for audio generation tasks. Diffusion transformers (DiTs) outperform traditional diffusion models with UNet backbones in both performance and flexibility [7], [8]. The hierarchical convolutional structure of UNet models presents scalability challenges, limiting their effectiveness in handling complex tasks like audio generation [9]. In contrast, DiTs [10], [11] leverage transformer architectures to better capture long-range temporal dependencies and intricate spectral patterns that are critical in audio generation. This makes them great for tasks like generating realistic instrument sounds [12], smooth soundscapes [13], and natural-sounding speech [14] with expressive tones. Models like Stable Audio [10] show how these systems can create high-quality audio clips with consistent timing and sound detail.

Despite their success across various generative tasks, DiTs face significant challenges due to their high computational requirements and increased storage demands [15], [16]. To address this, researchers have turned to model quantization, which reduces computation and memory demands by using lower bitwidths for weights and activations. Among these techniques, post-training quantization (PTQ) stands out as a practical and straightforward approach [17]. Unlike quantization-aware training (QAT), which requires retraining the entire model, PTQ uses a small dataset for quick calibration to adjust scale factors and minimize quantization errors. This makes PTQ particularly suitable for quantizing DiTs from 32-bit floating-point weights into 8-bit or 4-bit integers without the need for extensive computational resources. PTQ can also convert the activations (e.g. the input of a linear layer) from 32-bit float numbers to 8-bit integers. As a result, the matrix multiplications of both attention modules and linear layers could take place in the low-precision integer field, thus accelerating the inference process and reducing the memory footprint.

While most quantization research to date has focused on UNet-based diffusion models [18], [19]—particularly in text-to-audio generation tasks [20]—transformer-based diffusion models such as DiTs remain largely underexplored in the audio domain. This gap is notable given DiT's superior performance in audio generation [10]. Most PTQ methods for diffusion models rely on fixed-point quantization, which can introduce significant errors at lower precision, resulting in performance degradation. When these methods are applied to DiTs, two major challenges arise. First, certain channels within the model—often referred to as salient channels—can exhibit extremely large or small values compared to others [21]. This imbalance disrupts uniform scaling, causing substantial quantization errors. Second, the distribution of activations in DiTs changes significantly across different timesteps of the diffusion process. Early timesteps are dominated by noise [21], while later timesteps focus on refining fine-grained audio details, resulting in highly variable activation ranges throughout inference. As a result, a single, static quantization range is often insufficient to accommodate these variations, leading to cumulative errors and degraded generation quality. Recent work has begun to address quantization challenges, particularly for image generation. These methods target high-activation layers [20] and address activation variability via techniques like channel-wise salience balancing [21]. Building on this, PTQ4DM [22] uses timestep-aware calibration, Q-Diffusion [18] introduces split shortcuts for 4-bit quantization, APQ-DM [23] applies group-wise rounding, and PTQD [24] adds variance correction for mixed precision. Recent advances include SVDQuant's [25] low-rank outlier suppression and DiTAS's [26] layer-wise grid search strategy with temporal smoothing. Despite these advances, there remains limited insight into how DiTs behave specifically in audio generation tasks, whether the same issues arise, and how effectively these models can be quantized.

In this paper, we conduct a comprehensive study of PTQ strategies for audio DiTs. We analyze the behavior of a widely used audio generation DiT model (i.e, we look at activation and weights ranges and outliers), and we introduce two practical extensions tailored to audio DiTs. First, denoising-timestep-aware smoothing strategy based on SmoothQuant [27], which scales activations and weights individually for each timestep and channel, addressing the dynamic activation distributions inherent in diffusion models. Secondly, to mitigate degradation in generation performance, we assess integrating low-rank adaptation (LoRA) [28] modules into the quantized weights of the DiT model. Specifically, we apply singular value decomposition (SVD) to the smoothed and quantized weight matrices, decomposing them into a low-rank component and a residual. This decomposition allows us to compensate for quantization errors by isolating the residuals into trainable low-rank approximations. We investigate the effects of each technique, individually and in combination, across static and dynamic quantization regimes. Our results provide insights into which configurations best preserve the generation quality of audio DiTs, as measured by both objective metrics and human evaluations.

2. METHODOLOGY

For our analysis, we chose Stable Audio Open [29] because it is fully open-source and provides open access to the model's weights. Moreover, its architecture combines an autoencoder, T5-based text conditioning, and transformer-based diffusion, which is representative of modern DiTs for audio generation. Finally, the model is optimized for consumer GPUs, has strong community support, and has reproducible computational benchmarks, thus ideal for our study. Same as Stable Audio Open, we use AudioCaps [30] as benchmark.

Most DiTs, including Stable Audio's, are constructed from stacks of transformer blocks, each comprising self-attention layers and multilayer perceptron (MLP) modules [7], [31]. Within these blocks, both the feed-forward networks (FFNs) and the query-key-value (QKV) projection layers of self-attention are major contributors to computational cost. FFNs alone account for over 60% of model parameters and up to 70% of total FLOPs. Similarly, the QKV projections in self-attention require large linear transformations to compute the query, key, and value representations for each token, further increasing the computational and memory demands. As a result, both FFNs and QKV layers are critical targets for PTQ. We start by analyzing the input activations of both FFNs and QKV projections using forward passes on randomly selected prompts from the validation set, recording per-channel activation ranges to understand how activation values behave in audio DiTs. Figures 1 and 2 provide visual intuition for the design choices in our study.
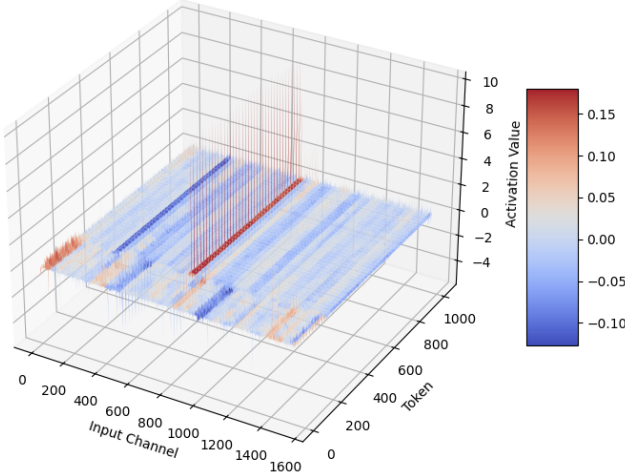

Fig. 1: Activation map at denoising timestep 50 for DiT Block 24, showing activation values across tokens and input channels.

First, Figure 1 shows a 3D visualization of the activation distributions at time step 50 for DiT Block 24, plotting activation values across both tokens and input channels. During our analysis, one major issue we observed was the large variation in activation values across input channels, particularly in the QKV projections of the self-attention layers and the FFN layers. The vertical spikes (both in red and blue, indicating positive and negative values) make clear that certain input channels yield significantly larger magnitudes—sometimes extreme outliers, while others hover near zero. As a result, channels with more moderate activations would suffer from elevated quantization error. Although quantization is typically performed on output channels as it is hardware-efficient, we observed that the uneven activation patterns across these input channels, together with the presence of large outliers, would severely skew the quantization parameters. Motivated by this

and following insights from prior work [26], we instead quantize activations taking into account input channels.
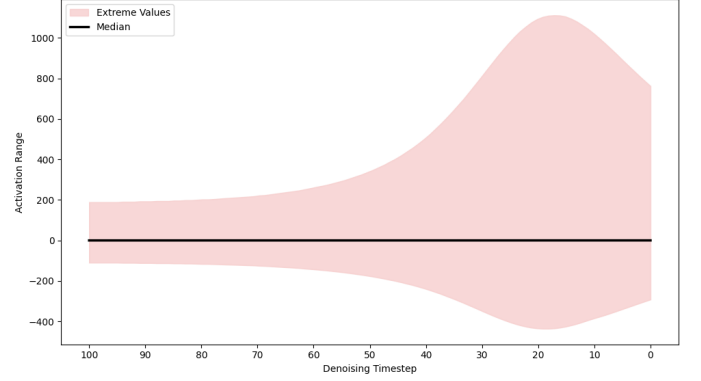

Fig. 2: Visualization of input activation range across denoising timesteps $(100 \rightarrow 0)$ for Block 1. The shaded region represents the full activation span (min to max), while the solid line denotes the median activation. As denoising progresses, the range of activations increases significantly, highlighting the emergence of outliers in later steps.

Second, we observed from Figure 2 how the activation values expand during the denoising process, with the horizontal axis representing the denoising timesteps and the vertical axis capturing the range of activation values. As the timesteps advance, the distribution widens and occasionally spans extreme magnitudes. Static quantization methods, which are designed around fixed activation ranges, would result in amplified errors at later timesteps.

Our objective is to tackle outliers in the activations across input channels, and the varying ranges of those activations over timesteps. For that we adapt SmoothQuant [27], usually applied in the context of large language models (LLMs), and introduce a per-input-channel, time-aware smoothing factor to reduce the impact of activation outliers. By storing the maximum activation values for each channel and timestep during the denoising process, we dynamically adjust quantization parameters to account for temporal and channel-wise variation.

Our extension is as follows: Consider a linear layer where $\mathbf{X}^{(t)} \in \mathbb{R}^{k \times n}$ denotes the activation matrix at timestep t, with k channels and n elements per channel. Let $\mathbf{W}$ be the corresponding weight matrix.

For each channel $j \in \{1, \ldots, k\}$, we record the maximum absolute activation:

$$\mathbf{X}_{\text{absmax},j}^{(t)} = \max\left(|\mathbf{X}_j^{(t)}|\right), \tag{1}$$

and the corresponding maximum absolute weight:

$$\mathbf{W}_{\text{absmax},j} = \max\left(|\mathbf{W}_j|\right). \tag{2}$$

Using these values, we define a per-channel smoothing factor:

$$\mathbf{s}_j^{(t)} = \frac{\left(\mathbf{X}_{\text{absmax},j}^{(t)}\right)^\alpha}{\left(\mathbf{W}_{\text{absmax},j}\right)^{1-\alpha}}, \quad \alpha \in [0, 1], \tag{3}$$

which balances the influence of activations and weights. A larger α results in stronger attenuation of large activation values, while a smaller α emphasizes weight scaling.

This smoothing is implemented by rescaling both activations and weights:

$$\hat{\mathbf{X}}_j^{(t)} = \frac{\mathbf{X}_j^{(t)}}{\mathbf{s}_j^{(t)}}, \qquad \hat{\mathbf{W}}_j = \mathbf{W}_j \cdot \mathbf{s}_j^{(t)}, \tag{4}$$

such that the resulting linear transformation remains algebraically identical:

$$\mathbf{Y} = \hat{\mathbf{X}}^{(t)}\hat{\mathbf{W}} = \left(\frac{\mathbf{X}^{(t)}}{\mathbf{s}^{(t)}}\right)\left(\mathbf{W} \cdot \mathbf{s}^{(t)}\right) = \mathbf{X}^{(t)}\mathbf{W}. \qquad (5)$$

Figure 3 shows the intuition behind this. The top-left panel shows the absolute activation values $|\mathbf{X}|$ before smoothing. Here, a single large outlier dominates the range, forcing the quantizer to reserve most of its dynamic range for rare, extreme values. This leads to low effective bits for the remaining, more common activation values, making them difficult to quantize precisely. Meanwhile, the top-right panel shows the corresponding weight distribution $|\mathbf{W}|$, which is smoother with fewer outliers and thus easier to quantize.

To address this imbalance, we compute a smoothing factor that rebalances the dynamic ranges between activations and weights, effectively reducing the impact of outliers during quantization. We call this method SmoothQuant Dynamic (SQD).
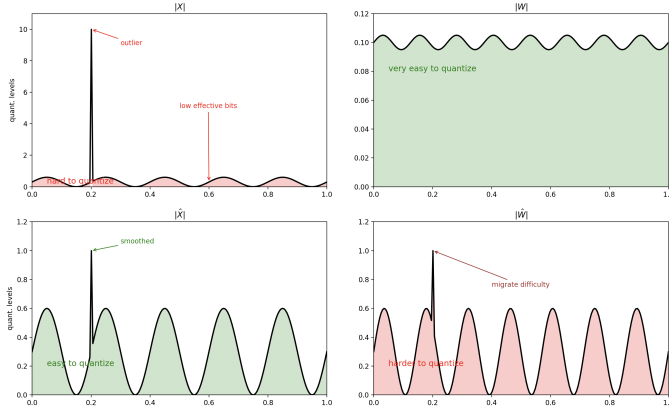

Fig. 3: Visualizing "easy vs. hard to quantize" regions and outliers. The spikes in activations ("outliers") lead to low effective bits for other channels, whereas flatter distributions ("smoothed") are more amenable to quantization.

Figure 3 (bottom) further illustrates how Eq. (3) flattens sharp activation peaks, yielding distributions that are easier to quantize. While smoothing makes $\hat{\mathbf{W}}$ more quantization-friendly, the finite-precision representation still introduces residual error. To mitigate this, we introduce a 16-bit low-rank branch. Intuitively, this low-rank branch is aimed at capturing the most important components of the quantization error, and correct them. The transformed weight matrix is decomposed into $\mathbf{AB}^\top$. Compared to direct 4/8-bit quantization, we first compute the low-rank branch in 16-bit precision and then approximate the residual in 4-bit or 8-bit quantization. As a result, the additional parameters and computational overhead for the low-rank branch are negligible. To find the low-rank branch, we first compute the residual:

$$\mathbf{E} = \mathbf{W} - \hat{\mathbf{W}}, \qquad \mathbf{E} \in \mathbb{R}^{k \times m}, \qquad (6)$$

then model the structured component of $\mathbf{E}$ using truncated SVD:

$$\mathbf{E} \approx \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^\top, \qquad r \ll \min(k, m), \qquad (7)$$

where $\mathbf{U}_r \in \mathbb{R}^{k \times r}$, $\mathbf{\Sigma}_r = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$, and $\mathbf{V}_r \in \mathbb{R}^{m \times r}$.

We then define:

$$\mathbf{A} = \mathbf{U}_r \mathbf{\Sigma}_r^{1/2}, \qquad \mathbf{B} = \mathbf{V}_r \mathbf{\Sigma}_r^{1/2}, \qquad (8)$$

so that $\mathbf{AB}^\top = \mathbf{U}_r \mathbf{\Sigma}_r \mathbf{V}_r^\top$. Both $\mathbf{A}$ and $\mathbf{B}$ are kept in FP16.

The final weights used at inference time are:

$$\tilde{\mathbf{W}} = \underbrace{\hat{\mathbf{W}}}_{\text{INT8/INT4 core}} + \underbrace{\mathbf{AB}^\top}_{\text{FP16 adaptor}}. \qquad (9)$$

Given an INT8/INT4-quantized input $\left\lfloor \mathbf{X}^{(t)}/\mathbf{x}_{\text{scale}} \right\rceil$, matrix multiplication with $\tilde{\mathbf{W}}$ is accumulated in FP16 or FP32 to preserve numerical precision.[1]

We also implement a lightweight variant called SmoothQuant Static (SQS). Unlike SQD, which adapts scales dynamically per timestep, this variant applies static quantization to both weights and activations based on precomputed statistics. During calibration, each layer tracks the per-input-channel running minimum and maximum of activations across denoising steps. After collecting these statistics, we compute a single global maximum per channel to derive the SmoothQuant scale and fold it into the weights. Activations are then quantized using the global min/max range, producing fixed scaling parameters. This requires no runtime adaptation or fine-tuning, making SQS efficient and low-latency.

3. EXPERIMENTAL DESIGN

This section details the experimental process for our quantized Stable Audio model on the audio generation task. Our methodology closely follows the evaluation protocol described in the original Stable Audio Open paper [29]. We use the pre-trained Stable Audio Open model as our full-precision baseline, operating at a 44.1 kHz sampling rate and generating 10-second audio clips.

For audio generation, we employ the DPM-Solver++ sampler with 100 steps, using classifier-free guidance (CFG) set to 7.0 to enhance output quality. Noise levels are managed with $\sigma_{\min} = 0.3$ and $\sigma_{\max} = 500$. The model sourced from Hugging Face, serves as the foundation for our experiments, upon which we apply various quantization techniques.

The evaluation is conducted using the AudioCaps evaluation dataset [32], which originally contains 979 YouTube audio segments, each paired with multiple captions. After filtering out inaccessible files, we retain 881 audio segments and 4,875 corresponding captions. These captions are used to generate 4,875 audio clips, mirroring the procedure in the Stable Audio Open paper. All experiments are performed on a single NVIDIA A100 GPU. To ensure format compatibility, the audio is peak-normalized, clipped, and converted to 16-bit PCM.

To maintain comparability with Stable Audio Open, we use three established evaluation metrics to thoroughly assess the quality and relevance of audio generated by our quantized Stable Audio model. The first metric, FD_{openl3}, compares the feature distributions of generated and reference audio. Lower FD_{openl3} scores indicate that the generated audio closely resembles real audio, reflecting high fidelity. The second metric, KL_{passt}, measures semantic similarity by comparing distributions of audio tags predicted by a pre-trained tagger. A lower KL_{passt} score means the generated audio captures the same semantic content as the reference, indicating strong alignment in meaning and content. The third metric, $CLAP_{\text{score}}$, evaluates how well the generated audio matches the provided text prompt by comparing embeddings of the audio and its caption. A higher $CLAP_{\text{score}}$ shows that the generated audio accurately reflects the intent and details of the input text.

To assess model efficiency, we compare the size of the model before and after applying quantization methods such as SmoothQuant and LoRA. Model size is measured by saving the state dictionary and recording the file size, with the original full-precision model occupying approximately 4,854 MB. We focus on two quantization configurations: W8A8 (8-bit weights and activations) and W4A8 (4-bit weights, 8-bit activations).

---

[1]Note that this accumulation in FP16/FP32 does not increase the model's memory footprint, which aligns with the primary goal of our work. While executing all operations in lower precision could further accelerate computation, this is beyond the scope of this paper and left for future work.

For PTQ, we use a calibration set of 50 randomly selected prompts. This set is used both for SmoothQuant calibration with the hyperparameter α set to 0.5 and for computing the SVD of the LoRA components. Our implementation applies per-output-channel symmetric quantization for weights and per-input-channel symmetric quantization for activations. Experiments are conducted using both W8A8 and W4A8 configurations to systematically evaluate the trade-offs between compression and generation quality.

4. RESULTS AND DISCUSSION

We establish the baseline using the original, full-precision model. During preliminary experiments, we observed that evaluation metrics varied substantially based on the random seed, often diverging from the originally reported values in earlier studies. To ensure consistency and fairness, we systematically tested multiple seeds and ultimately selected seed = 1000, which yielded results close to those reported and high-quality generations. Our full-precision results achieve a CLAP Score of 0.3009, KL_{passt} of 2.17, and FDopenl3 of 87.02, and a best-case generation latency of $\sim 11.3\,\mathrm{s}$. We use this as the baseline, but still include results reported in the original paper in our table.

Table 1: Performance comparison of full-precision and quantized Stable Audio models using SmoothQuant and LoRA for both dynamic (i.e. channel- and step-dependent) and static cases (i.e. single value for all channels and steps). SQD = SmoothQuant Dynamic; SQS = SmoothQuant Static. LoRA denotes low-rank adaptation. ↑ indicates higher is better; ↓ indicates lower is better. Best results in **bold**, second best underlined.

Precision	Variant	CLAP ↑	KL_{passt} ↓	FD_{openl3} ↓	Size (GB) ↓
FP32	Reported	0.2900	2.14	**78.24**	–
FP32	Baseline	0.3009	2.17	87.02	4.85
W8A8	SQD	<u>0.3021</u>	2.158	86.35	<u>1.65</u>
W8A8	SQS	0.2934	2.144	<u>80.57</u>	<u>1.65</u>
W8A8	SQD+LoRA	**0.3033**	2.153	85.70	1.71
W4A4	SQD	0.2901	**2.039**	82.57	**1.03**
W4A4	SQS	0.2014	2.780	224.7	**1.03**
W4A4	SQD+LoRA	0.2829	<u>2.096</u>	85.85	1.17

We evaluate three quantization strategies under two precision settings: W8A8 and W4A8. The quantization strategies are SmoothQuant Dynamic (SQD) with and without low-rank adaptation (+LoRA), and SmoothQuant Static (SQS). Please see Section 2 for details about these methods. Results are shown in Table 1.

We found that SQD models closely match or even surpass our full-precision baseline across objective metrics (CLAP, KL_{passt}, and FD_{openl3}) for the two precision configurations. This shows that dynamic calibration effectively handles activation outliers at each timestep, with minimal performance loss after quantization.

For W8A8, LoRA consistently boosted metrics, narrowing any remaining gap to the FP32 baseline. At W4A8, however, LoRA did not yield consistent improvements. Given that the quantization error is often too severe in this setting, especially in the presence of activation outliers, a low-rank additive correction (like LoRA) falls short.

We noted that the static approach performs competitively at W8A8, offering a strong trade-off between simplicity and quality. However, in the more aggressive W4A8 setting, static quantization results in significant degradation. This suggests that in more aggressive quantization settings, there are bigger advantages to adjusting dynamically to activation outliers.

A key limitation of our SQD approach compared to SQS is its slower inference speed ($\sim 35.6\,\mathrm{s}$ vs. $\sim 11.6\,\mathrm{s}$), which is primarily due to the overhead of maintaining scaling factors that are specific to each timestep and input channel, as well as the need for dynamic
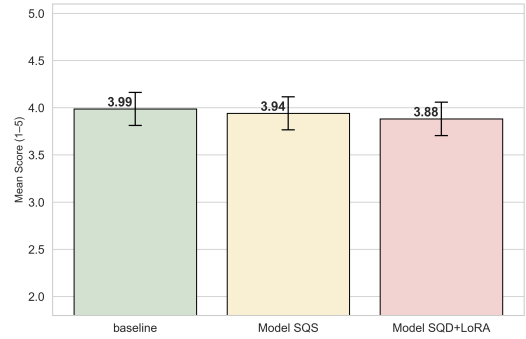

Fig. 4: Subjective evaluation of mean user ratings (1–5 scale) in W8A8 for the full-precision baseline (baseline), the fastest variant (Model SQS), and the best-performing model (Model SQD+LoRA).

computation during inference. This can be mitigated through caching, pruning unused quantization paths, or integrating fast integer-aware operators—remains, and remains an avenue for future work.

We conducted a subjective evaluation to complement our objective metrics. Specifically, we compared our best-performing model (SQD + LoRA) with the fastest configuration (SQS), both in W8A8 precision, alongside the original full-precision baseline. To ensure diversity in auditory content, we selected five prompts spanning various sound classes and constructed a 15-question survey by randomly sampling these prompts across the three model variants. The survey was distributed to 20 participants. The participants were asked to rate each audio sample on a 1–5 scale based on perceived alignment with the input prompt. These qualitative judgments were then aggregated into a single composite score per model to quantify perceptual performance.

Based on 100 ratings per model (300 in total), the full-precision baseline attained the highest mean score of 3.99. The W8A8-quantized SQS variant followed closely at 3.94, while the SQD + LoRA configuration achieved 3.88. This suggests that both quantized variants preserve perceptual quality remarkably well. In particular, the quantized SQS variant maintains perceptual fidelity nearly indistinguishable from the full-precision baseline. Meanwhile, the SQD + LoRA model achieves the highest CLAP score and remains competitive on other, but exhibits slightly lower subjective ratings—suggesting that LoRA fine-tuning may enhance objective alignment more than perceptual quality.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we conducted a comprehensive study of PTQ strategies for audio DiTs, with a focus on the trade-offs between static and dynamic calibration. We introduced two practical extensions: denoising-timestep-aware smoothing and LoRA to compensate for residual weight errors. Our results show that SQD, with or without LoRA, preserves generation quality across both 8-bit and 4-bit settings, closely matching the full-precision baseline on objective metrics. While static quantization works well at 8 bits, it deteriorates at 4 bits, underscoring the need for dynamic calibration. LoRA improves 8-bit performance but has a limited impact at lower precision. A key limitation of SQD is slower inference, driven by the cost of dynamic scaling. Subjective evaluations confirm that quantized models remain perceptually close to the baseline. Future work will explore faster implementations of dynamic quantization and full low-precision execution to further accelerate inference of DiT audio models.

6. ACKNOWLEDGMENTS

REFERENCES

[1] H. Flores García, O. Nieto, J. Salamon, B. Pardo, and P. Seetharaman, "Sketch2sound: Controllable audio generation via time-varying signals and sonic imitations," *arXiv preprint arXiv:2412.08550*, 2024. [Online]. Available: https://arxiv.org/abs/2412.08550

[2] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "AudioLDM: Text-to-audio generation with latent diffusion models," in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 21 450–21 474. [Online]. Available: https://proceedings.mlr.press/v202/liu23f.html

[3] M. Levy, B. D. Giorgi, F. Weers, A. Katharopoulos, and T. Nickson, "Controllable music production with diffusion models and guidance gradients," in *NeurIPS*, 2023. [Online]. Available: http://arxiv.org/abs/2311.00613

[4] Q. Huang, D. S. Park, T. Wang, T. I. Denk, A. Ly, N. Chen, Z. Zhang, Z. Zhang, J. Yu, C. Frank, J. Engel, Q. V. Le, W. Chan, and W. Han, "Noise2Music: Text-conditioned music generation with diffusion models," *arXiv preprint arXiv:2302.03917*, 2023. [Online]. Available: https://arxiv.org/abs/2302.03917

[5] P.-L. W. L. Suckrow, C. J. Weber, and S. Rothe, "Diffusion-based sound synthesis in music production," in *Proceedings of the 12th ACM SIGPLAN International Workshop on Functional Art, Music, Modelling, and Design*, ser. FARM 2024. New York, NY, USA: Association for Computing Machinery, 2024, p. 55–64. [Online]. Available: https://doi.org/10.1145/3677996.3678289

[6] G. Zhu, Y. Wen, M.-A. Carbonneau, and Z. Duan, "Edmsound: Spectrogram based diffusion models for efficient and high-quality audio synthesis," 2023. [Online]. Available: https://arxiv.org/abs/2311.08667

[7] W. Peebles and S. Xie, "Scalable diffusion models with transformers," 2023. [Online]. Available: https://arxiv.org/abs/2212.09748

[8] F. Bao, S. Nie, K. Xue, Y. Cao, C. Li, H. Su, and J. Zhu, "All are worth words: A ViT backbone for diffusion models," 2023. [Online]. Available: https://arxiv.org/abs/2209.12152

[9] N. Ding, J. Han, Y. Tian, C. Xu, K. Han, and Y. Tang, "Post-training quantization for diffusion transformer via hierarchical timestep grouping," 2025. [Online]. Available: https://arxiv.org/abs/2503.06930

[10] Z. Evans, C. Carr, J. Taylor, S. H. Hawley, and J. Pons, "Fast timing-conditioned latent audio diffusion," in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24. JMLR.org, 2024.

[11] Z. Tian, Y. Jin, Z. Liu, R. Yuan, X. Tan, Q. Chen, W. Xue, and Y. Guo, "AudioX: Diffusion transformer for anything-to-audio generation," 2025. [Online]. Available: https://arxiv.org/abs/2503.10522

[12] S. Li and Y. Sung, "MelodyDiffusion: Chord-conditioned melody generation using a transformer-based diffusion model," *Mathematics*, vol. 11, p. 1915, 04 2023.

[13] M. Heydari, M. Souden, B. Conejo, and J. Atkins, "ImmerseDiffusion: A generative spatial audio latent diffusion model," in *ICASSP*, 2025. [Online]. Available: https://arxiv.org/abs/2410.14945

[14] H. J. Park, J. S. Kim, W. Shin, and S. W. Han, "DEX-TTS: Diffusion-based expressive text-to-speech with style modeling on time variability," *arXiv preprint arXiv:2406.19135*, 2024.

[15] J. Lou, W. Luo, Y. Liu, B. Li, X. Ding, W. Hu, J. Cao, Y. Li, and C. Ma, "Token caching for diffusion transformer acceleration," 2024. [Online]. Available: https://arxiv.org/abs/2409.18523

[16] X. Ma, G. Fang, and X. Wang, "Deepcache: Accelerating diffusion models for free," 2023. [Online]. Available: https://arxiv.org/abs/2312.00858

[17] J. Zhang, Y. Zhou, and R. Saab, "Post-training quantization for neural networks with provable guarantees," 2023. [Online]. Available: https://arxiv.org/abs/2201.11113

[18] X. Li, Y. Liu, L. Lian, H. Yang, Z. Dong, D. Kang, S. Zhang, and K. Keutzer, "Q-diffusion: Quantizing diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 17 535–17 545. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2023/html/Li_Q-Diffusion_Quantizing_Diffusion_Models_ICCV_2023_paper.html

[19] R. Zhan, J. Chen, H. Yang, D. Huang, and Y. Wang, "Temporal distribution-aware quantization for diffusion models," in *International Conference on Learning Representations (ICLR)*, 2025, under review. [Online]. Available: https://openreview.net/forum?id=lF5U9jTdyq

[20] J. Vora, A. Krishnan, N. Bouacida, P. R. Shankar, and P. Mohapatra, "PTQ4ADM: Post-training quantization for efficient text conditional audio diffusion models," *ArXiv*, vol. abs/2409.13894, 2024. [Online]. Available: https://arxiv.org/abs/2409.13894

[21] J. Wu, H. Wang, Y. Shang, M. Shah, and Y. Yan, "PTQ4DiT: Post-training quantization for diffusion transformers," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024, poster. [Online]. Available: https://neurips.cc/virtual/2024/poster/95445

[22] Y. Shang, Z. Yuan, B. Xie, B. Wu, and Y. Yan, "Post-training quantization on diffusion models," in *CVPR*, 2023.

[23] C. Wang, Z. Wang, X. Xu, Y. Tang, J. Zhou, and J. Lu, "Towards accurate post-training quantization for diffusion models," 2024. [Online]. Available: https://arxiv.org/abs/2305.18723

[24] Y. He, L. Liu, J. Liu, W. Wu, H. Zhou, and B. Zhuang, "PTQD: Accurate post-training quantization for diffusion models," 2023. [Online]. Available: https://arxiv.org/abs/2305.10657

[25] 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," 2025. [Online]. Available: https://arxiv.org/abs/2411.05007

[26] Z. Dong and S. Q. Zhang, "DiTAS: Quantizing diffusion transformers via enhanced activation smoothing," *arXiv preprint arXiv:2409.07756*, 2024.

[27] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, "Smoothquant: Accurate and efficient post-training quantization for large language models," 2024. [Online]. Available: https://arxiv.org/abs/2211.10438

[28] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021. [Online]. Available: https://arxiv.org/abs/2106.09685

[29] Z. Evans, J. D. Parker, C. Carr, Z. Zukowski, J. Taylor, and J. Pons, "Stable audio open," 2024. [Online]. Available: https://arxiv.org/abs/2407.14358

[30] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 119–132. [Online]. Available: https://aclanthology.org/N19-1011/

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023. [Online]. Available: https://arxiv.org/abs/1706.03762

[32] C. D. Kim, B. Kim, H. Lee, and G. Kim, "Audiocaps: Generating captions for audios in the wild," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 2019, pp. 119–132.