DM-Tune: Quantizing Diffusion Models with Mixture-of-Gaussian Guided Noise Tuning

Pouya Haghi*, Ali Falahati*, Zahra Azad*, Chunshu Wu*, Ruibing Song*, Chuan Liu*, Ang Li†‡, Tong Geng*

*University of Rochester, Rochester, NY
†Pacific Northwest National Laboratory, Richland, WA
‡University of Washington, Seattle, WA
*{phaghi, afalahat, zazad, cwu88, rsong10, cliu81, tgeng}@ur.rochester.edu, †ang.li@pnnl.gov

2025 62nd ACM/IEEE Design Automation Conference (DAC) | 979-8-3315-0304-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/DAC63849.2025.11132501

Abstract*—Diffusion models have become essential generative tools for tasks such as image generation, video creation, and inpainting, but their high computational and memory demands pose challenges for efficient deployment. Contrary to the traditional belief that full-precision computation ensures optimal image quality, we demonstrate that a fine-grained mixed-precision strategy can surpass full-precision models in terms of image quality, diversity, and text-to-image alignment. However, directly implementing such strategies can lead to increased complexity and reduced runtime performance due to the overheads of managing multiple precision formats and casting operations. To address this, we introduce *DM-Tune*, which replaces complex mixed-precision quantization with a unified low-precision format, supplemented by noise-tuning, to improve both image generation quality and runtime efficiency. The proposed noise-tuning mechanism is a type of fine-tuning that reconstructs the mixed-precision output by learning adjustable noise through a parameterized nonlinear function consisting of Gaussian and linear components. Key steps in our framework include identifying sensitive layers for quantization, modeling quantization noise, and optimizing runtime with custom low-precision GPU kernels that support efficient noise-tuning. Experimental results across various diffusion models and datasets demonstrate that DM-Tune not only significantly improves runtime but also enhances diversity, quality, and text-to-image alignment compared to FP32, FP8, and state-of-the-art mixed-precision methods. Our approach is broadly applicable and lays a solid foundation for simplifying complex mixed-precision strategies at minimal cost.

***Index Terms*—Diffusion Models, Quantization, Mixed-precision, GPU.**

I. Introduction

Diffusion models serve as potent tools for tasks like image generation, video creation, and inpainting, yet they demand significant computational and memory resources [1]–[5]. For instance, generating a single image using Stable Diffusion XL (SDXL) [6], which has approximately 10 billion parameters, requires over two minutes on an A100 GPU, highlighting the urgency for efficiency enhancements.

Recent AI hardware advancements [7], especially GPUs, often rely on low-precision computation to improve performance. For instance, Nvidia's upcoming Blackwell GPUs support a variety of floating-point (FP) formats (from FP16, and BF16, down to FP8, and FP4), yielding a vast search space for efficient execution of deep learning models.

Quantization leverages low-precision hardware to reduce memory and computation costs, but fully quantizing diffusion models to low precision (e.g., 8-bit) often results in low-quality images. Previous research explored mixed-precision strategies, such as intra-layer and timestep-aware methods [8]–[10]. However, these approaches face two main challenges.

1) Full-precision fallacy: It is traditionally believed that full-precision computation offers the highest accuracy. Our research shows that an FP-based mixed-precision (FP-MP) quantization strategy can outperform full-precision in image generation quality, providing more detail and better prompt alignment. Fig. 1 (a) compares images from FP32 and FP-MP (BF16 and FP8) using a Stable Diffusion model. Diffusion models absorb quantization noise into their inherent noise, potentially improving metrics. Unlike previous integer-based methods, our FP approach enhances performance. We compare with PTQ4DM [9], finding that integer-based strategies produce lower quality images. *2) Mixed-precision overhead:* However, mixed-precision can be slower than full-precision due to complex strategies and casting overheads, with low-precision resources underutilized.

To solve these problems, we propose *DM-Tune*, a framework that first finds an efficient FP-MP strategy that outperforms full-precision by progressively introducing **controlled noise**. It then replaces the FP-MP model with a **unified low-precision** model and a **noise-tuning** head to optimize quality and speed. Noise-tuning fine-tunes the model by adding a guided noise head, with minimal overhead, as only the head's parameters are trained and applied once before inference.

To achieve this goal, three challenges must be addressed. First, the vast search space for the optimal FP-MP strategy. Second, effectively expressing the noise-tuning operator. Third, noise-tuning slows the model inference due to memory demands. To resolve the first challenge, we identify sensitive layers and then propose two novel techniques to enhance the quality: *prompt-aware* and *timestep-aware quantization*. Second, we recognize that multiple overlapping Gaussians are needed as nonlinear functions to recover FP-MP output. Finally, we provide a highly optimized GPU kernel that fuses matrix multiplication in low-precision with nonlinear Gaussian terms. This approach reduces runtime and enhances image quality over state-of-the-art (SOTA) and full-precision models (Fig. 1 (b) and (c)). Our main contributions are as follows:

- We show that a novel mixed-precision strategy can

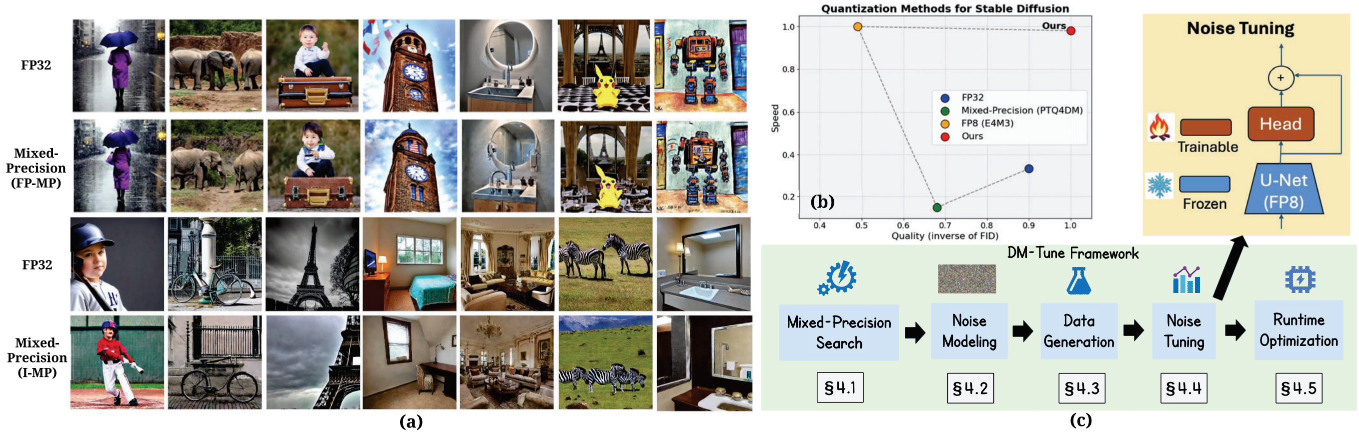

Fig. 1. (a) Generated images in FP32, FP-based mixed-precision (mixed BF16 and FP8), and integer-based mixed precision (Q-Diffusion) using a fixed seed. FP-MP offers greater detail and better prompt alignment compared to FP32. For example, in the prompt ”*a view of a multi-tiered clock tower with a US flag on the top*,” FP32 misplaces the flags, leaving them suspended in the sky, whereas FP-MP correctly positions a flag at the top of the tower. In contrast, I-MP struggles with image regeneration, particularly when generating human faces. (b) Comparing image quality and speed of our approach with PTQ4DM, FP32, and FP8. (c) Overview of DM-Tune.

outperform full-precision image generation quality and diversity for different types of diffusion models.

- We introduce noise-tuning, a data-free technique for running models in low-precision at high speed with mixed-precision quality.
- We develop highly optimized GPU kernels that fuse low-precision matrix multiplication with nonlinear functions.
- Experimental results show that our approach improves the runtime of prior art by 5.2× while improving image generation quality and diversity.

II. Background

Diffusion Models: Diffusion models generate images using a Markov chain process. Initially, a forward diffusion process adds Gaussian noise to the data $x_0 \sim q(\mathbf{x})$ over T steps, resulting in progressively noisier samples $\mathbf{x}_1, \ldots, \mathbf{x}_T$:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I}) \tag{1}$$

Here, $\beta_t \in (0,1)$ is a variance schedule that determines the intensity of Gaussian noise at each step. As $T \to \infty$, $\mathbf{x}_T$ converges to an isotropic Gaussian distribution.

The backward process removes noise from a sample drawn from the Gaussian noise input $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$ to generate high-fidelity images. Since the actual reverse conditional distribution $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ is unknown, diffusion models sample from a learned conditional distribution:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\mu}_\theta(\mathbf{x}_t), \Sigma_\theta(\mathbf{x}_t, t)) \tag{2}$$

$$\tilde{\mu}_\theta(\mathbf{x}_t) = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right) \tag{3}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$. The variance $\Sigma_\theta(\mathbf{x}_t, t)$ can either be reparameterized or set to a constant schedule σ_t. When using a constant schedule, $\mathbf{x}_{t-1}$ is given by:

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right) + \sigma_t \mathbf{z} \tag{4}$$

where $\epsilon_\theta(\mathbf{x}_t, t)$ is the noise estimation model output at timestep *t*. The U-Net architecture [11] is predominantly used in designing the noise estimation model. For further details, we refer readers to [12]. This work focuses on quantization of the U-Net during the inference.

Evaluation of Diffusion Models: Evaluating diffusion models differs from other deep learning models due to potential biases and the inadequacy of single metrics like accuracy. Issues such as memorization and mode collapse affect evaluation [13]. We address these by using the *DINOv2-ViT* model as a feature extractor, rather than *Inception-V3*, and employ a variety of metrics. We measure performance with Fréchet Inception Distance (FID), Kernel Distance (KD), and Sliced-Wasserstein Distance (SW) [14]. We also evaluate prompt alignment with CLIP, and assess image quality, diversity, and authenticity using precision, density, recall, coverage, and authenticity metrics [15]. Lower scores are better for FID, KD, and SW, while higher scores are better for the other metrics.

III. Motivation

We summarize the key motivations of this work.

1) Mixed-precision outperforms full-precision: Contrary to traditional belief, FP-MP can surpass full-precision in diffusion models for two reasons. First, the stochastic nature of diffusion models means some seeds yield better results. Second, quantization noise merges with scheduler noise, beneficially altering the diffusion path. We can demonstrate this by rewriting (4) for the quantized model:

$$\begin{aligned}\mathbf{x}'_{t-1} &= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}'_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\left(\epsilon_\theta(\mathbf{x}'_t, t) + \Delta\epsilon_\theta(\mathbf{x}'_t, t)\right)\right) \\ &\quad + \sigma_t \mathbf{z} \\ &= \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}'_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}'_t, t)\right) \\ &\quad - \frac{\beta_t}{\sqrt{\alpha_t(1-\bar{\alpha}_t)}}\Delta\epsilon_\theta(\mathbf{x}'_t, t) + \sigma_t \mathbf{z}\end{aligned} \tag{5}$$

where $\mathbf{x}'_t$ and $\Delta\epsilon_\theta(\mathbf{x}'_t, t)$ are the quantized input and U-Net output quantization error at timestep *t*, respectively. We find mixed-precision benefits unique to floating-point formats (FP-MP), as integer-based mixed precision (I-MP) shows little improvement in image generation due to limited dynamic range and clipping. Thus, maintaining quality with integer quantization requires a complex mixed-precision strategy.

2) Mixed-precision slows down: Mixed-precision designs (I-MP and FP-MP) can be slower than full-precision due to

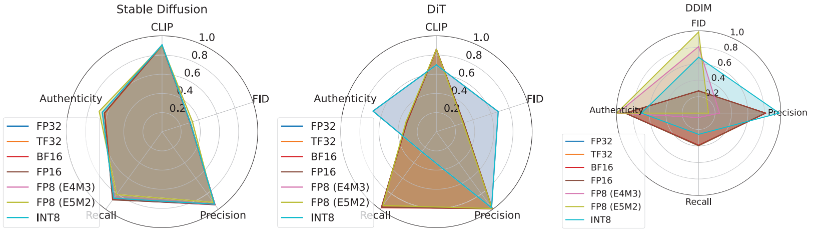

Fig. 2. Evaluating three diffusion models for different data formats.

complex quantization strategies. Therefore, GPUs often rely on unified low precision computation to enhance performance. We found that using unified low-precision for the entire U-Net while adding a nonlinear function with a negligible number of parameters is enough to recover FP-MP output. Our proposed FP-MP will be introduced at the end of Sec. IV-A.

IV. DM-Tune

In this section, we describe the components of DM-Tune framework. Our methodology provides a generalized and scalable solution that can be adapted to different models, datasets, and quantization techniques.

A. *Mixed-Precision Search*

In this step, we first select two precisions out of all supported data formats in GPUs. Subsequently, we provide our FP-MP quantization methodology to outperform full-precision.

1) Precision Selection: GPU vendors now support multiple data formats like FP32, FP16, BF16, FP8 (E4M3), and FP8 (E5M2). We simplify precision selection to two levels: *low* and *high*. Through experiments on three diffusion models (see Fig. 2), we find 16-bit quantization offers a significant speedup with comparable performance to 32-bit. BF16 generally outperforms FP16, making it our choice for *high-precision*. For *low-precision*, FP8 (E4M3) is preferred over FP8 (E5M2) for better performance since reducing exponent bits too much degrades image quality, while reducing mantissa bits enhances diversity. However, a 2-bit mantissa in 8-bit formats is insufficient.

2) Search Space Reduction: The challenge of mixed-precision search arises from the vast search space: with two formats (BF16, FP8), T timesteps, and L layers, it results in $2^{(T \times L)}$ possibilities. To address this, we reduce the space by classifying layers as either sensitive or insensitive. Sensitive layers require high precision to maintain model quality, while insensitive layers can be quantized to low precision without significant performance loss.

3) Sensitivity Criteria: To identify sensitive layers, we use two criteria: *(1) Range*: Layers with large ranges risk errors from value clipping. *(2) Standard Deviation (STD)*: High STD layers face significant quantization errors due to low precision. We sort layers by these criteria to guide quantization, as detailed in Algorithm 1.

4) Selecting Cutoff Point: After sorting layers by sensitivity, we determine the cutoff between *high* and *low* precision using a *binary search*. Evaluating each point directly is too time-consuming, so we start at the midpoint of the sensitivity list, checking model performance. If quality is below the threshold, we check the midpoint of the upper half; otherwise, the lower half. This continues until narrowed to one layer. Fig. 4 shows cutoff points for different models: DiT needs few high-precision layers, while DDIM requires most. Although this approach helps us achieve near-full-precision quality, it alone may not be sufficient to surpass it.

Algorithm 1 DM-Tune Algorithm for Identifying Sensitive Layers

```
Input: Calibration dataset, MAX_FLT (maximum allowable value). Output: sens_list
sens_list ← [], Shuffle the calibration dataset.
for each batch of samples in the calibration dataset do
    Randomly select a seed.
    Calculate the running average of overflow ratios for activation layers, storing in of_layers.
    Calculate the running average of std for activation layers, storing in std_layers.
end for
Sort of_layers by ratio, std_layers by std. i ← 0
while of_layers[i].ratio $\neq$ 0 do
    Push of_layers[i].layer to sens_list, i ← i + 1
end while
for i = 0 to size(std_layers) - 1 do
    if std_layers[i].layer is not in sens_list then
        Push std_layers[i].layer to sens_list.
    end if
end for
return sens_list
```

5) Surpassing Full-Precision: To outperform full-precision, we introduce two additional techniques:

(1) Prompt-aware quantization: conditional diffusion models often employ classifier-free guidance [16], which requires the model to process two inputs: one with the given prompt and another with a null prompt. We propose to quantize only the path associated with the input prompt to low-precision, while maintaining the null prompt path in high-precision. This selective approach adds controlled noise that can improve the quality of the output image.

(2) Timestep-aware quantization: We quantize insensitive layers to low precision during the early timesteps (first 80%) and maintain high precision in the final steps. This helps the model recover from early noise, producing sharper images, as final timestep precision is crucial to avoid blurriness.

By following the five steps outlined in this section, we achieve our proposed FP-MP design, which not only approaches the quality of full-precision but also exceeds it. To summarize, only the portion of the activation tensors that receives the prompt during early timesteps for insensitive layers are quantized to FP8 (E4M3), while the rest remain

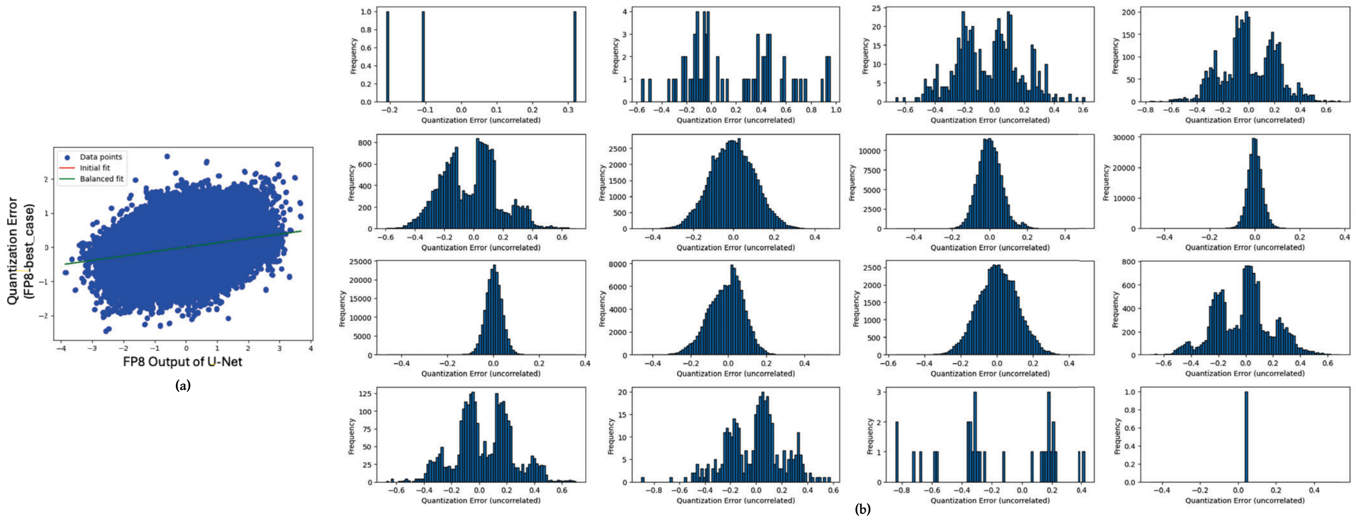

Fig. 3. (a) U-Net output quantization noise at timestep=12 for Stable Diffusion model using MS-COCO dataset: it is modeled with a correlated (linear) and an uncorrelated (nonlinear) component. (b) Distribution of the uncorrelated component for different ranges of U-Net output.

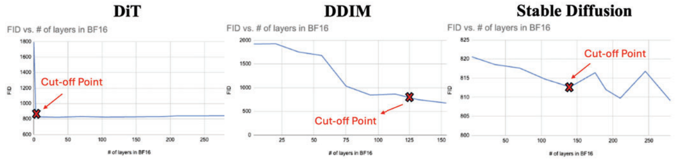

Fig. 4. Selecting cut-off point between high and low precision for different models: X-axis is the number of layers quantized to high-precision (the rest of the layers are quantized to low-precision) and Y-axis is FID (lower is better).

in BF16. For the weights, all sensitive (insensitive) layers are quantized to BF16 (FP8).

B. Noise Modeling

The goal of noise-tuning is to reconstruct FP-MP using low precision (FP8) with adjustable noise. We aim to correlate the U-Net low-precision output with the quantization error from transitioning to unified low precision, minimizing added parameters. Previous work [10] showed a linear correlation between the error and U-Net output for INT8 quantization.

We profile FP quantization error using the Stable Diffusion model and MS-COCO dataset. Fig. 3 (a) illustrates a *nonlinear* relationship between U-Net output in low-precision and quantization error, unlike integer quantization. The error comprises a correlated (linear) and an uncorrelated component (distance from the line). Fig. 3 (b) shows the uncorrelated component's distribution across U-Net output ranges, modeled by three overlapping Gaussians. Thus, we parameterize each Gaussian with trainable tensors that represent the mean, variance, and scaling of each distribution. We formulate the quantization noise as:

$$\mathbf{x}_{t-1,MP} = \frac{1}{\sqrt{\alpha_t}}\mathbf{x}_{t,MP} - \frac{\beta_t}{\sqrt{\alpha_t(1-\bar{\alpha}_t)}}\epsilon_{\theta,LP}(\mathbf{x}_{t,MP},t) - \frac{\beta_t}{\sqrt{\alpha_t(1-\bar{\alpha}_t)}}\Delta\epsilon_\theta(\mathbf{x}_{t,MP},t) + \sigma_t\mathbf{z} \tag{6}$$

$$\Delta\epsilon_\theta(\mathbf{x}_{t,MP},t) = P_{t,0}\cdot\epsilon_{\theta,LP}(\mathbf{x}_{t,MP},t) + \sum_{i=0}^{2} P_{t,1+3\cdot i}\cdot E$$
$$\text{where } E = \exp\left(-\frac{1}{2}\left(\frac{P_{t,2+3\cdot i}-\epsilon_{\theta,LP}(\mathbf{x}_{t,MP},t)}{P_{t,3+3\cdot i}}\right)^2\right) \tag{7}$$

where $\mathbf{x}_{t,MP}$, $\mathbf{z}$, $\epsilon_{\theta,LP}$, $\Delta\epsilon_\theta$ represent data sample at timestep t with FP-MP, a sample from distribution $\mathcal{N}(0,\mathbf{I})$, U-Net output in low-precision, and U-Net output quantization error (low-precision vs. FP-MP). α_t, β_t, and σ_t are hyperparamters. $P_{t,j} \in \mathbb{R}^{C\times H\times W}$ ($j \in \{0,1,\ldots,9\}$) is j^{th} trainable parameter at timestep t.

C. Data Generation

We generate training data for noise-tuning by curating input prompts. For conditional models, prompts are sampled from the dataset; The FP-MP design outperforms full-precision for conditional models, so we use FP-MP U-Net output as ground truth. For unconditional models, full-precision U-Net output is used, as FP-MP is less effective without prompts.

D. Noise-Tuning

The noise-tuning phase automates fine-tuning of U-Net head noise parameters using data-driven learning with ground truth from Sec. IV-C. Only the new parameters are trained, keeping the rest of the model frozen to minimize overfitting and reduce computational load. Training runs for a set number of epochs, halting if improvements plateau.

E. Runtime Optimization

The nonlinear function in noise-tuning is memory-bound, limiting DM-Tune's speed. Recent lower-bit quantization exacerbates this issue. To address this, we optimize a GPU kernel by fusing low-precision matrix multiplication with Gaussian terms, implementing FP8 matrix multiplication using CUTLASS[1] and high-throughput tensor cores.

[1] https://github.com/NVIDIA/cutlass

TABLE I
COMPARING DM-TUNE PERFORMANCE WITH DIFFERENT FORMATS FOR STABLE DIFFUSION, DIT, AND DDIM MODELS. NT IS NOISE-TUNING.

Model/Dataset	Method	FID ↓	Precision ↑	Recall ↑	Authenticity ↑	CLIP ↑	Density ↑	Coverage ↑	KD ↓	SW-approx ↓
Stable Diffusion (MS COCO 512×512)	FP32 (Baseline)	760.9	0.92	0.87	62.79	31.44	0.91	0.86	0.077	0.15
	FP8	776.44	0.92	0.86	67.18	31.56	0.91	0.88	0.091	0.16
	INT8	3993.55	0.0	0.0	100.0	19.16	0.0	0.0	27.62	1.73
	Q-Diffusion (W8A8)	757.51	0.92	0.87	63.14	31.40	0.88	0.86	0.077	0.15
	Q-Diffusion (W4A8)	768.90	0.92	0.83	63.57	31.27	0.92	0.87	0.085	0.16
	Ours (w/o NT)	**741.51**	**0.94**	0.86	60.94	**31.58**	**0.95**	**0.89**	**0.070**	**0.14**
	Ours (w/ NT)	**737.89**	0.91	**0.89**	**64.31**	31.32	**0.94**	**0.92**	**0.075**	**0.14**
DiT (ImageNet 256×256)	FP32 (Baseline)	721.58	0.99	0.97	30.78	30.10	0.89	0.98	0.045	0.14
	FP8	1626.56	0.98	0.31	70.41	24.31	0.75	0.91	1.88	0.63
	INT8	4300.40	0.0	0.0	100.0	19.16	0.0	0.0	32.1	1.91
	Ours (w/o NT)	**704.74**	0.99	0.97	**33.77**	29.91	**0.94**	**0.99**	0.045	**0.13**
	Ours (w/ NT)	**701.23**	0.98	0.97	**32.94**	**30.87**	**0.99**	0.96	**0.041**	**0.12**
DDIM (CelebAHQ 256×256)	FP32 (Baseline)	659.35	0.85	0.39	89.55	N/A	0.71	0.55	2.16	0.48
	FP8	1971.28	0.28	0.02	98.24	N/A	0.08	0.05	8.16	1.02
	INT8	4379.36	0.0	0.0	100.0	N/A	0.0	0.0	40.0	2.07
	Ours (w/o NT)	750.16	0.80	0.29	**91.02**	N/A	0.54	0.45	2.41	0.51
	Ours (w/ NT)	**653.98**	0.82	**0.44**	**90.14**	N/A	0.66	**0.58**	2.18	**0.45**

The noise-tuning head performs matrix multiplication, casts outputs to full-precision, adds Gaussian terms, and casts back to FP8 using intrinsic functions. We apply three optimizations to further improve the performance: **O1**: Replace (7) with lookup tables (LUTs) in shared memory. Based on the profiling results (i.e., Fig. 3 (a)), we set a range for the values of $\epsilon_{\theta,LP}(\mathbf{x}_{t,MP}, t)$ and allocate 32 KB shared memory. We calculate the LUT index based on the value of $\epsilon_{\theta,LP}(\mathbf{x}_{t,MP}, t)$. **O2**: Overlap exponential calculations with data loading to enhance efficiency. We use rolling prefetch; prefetching one element and calculating exponentials for another element. **O3**: Use vector instructions to improve instruction-level parallelism and throughput. We found that grouping eight FP8 elements leads to the highest throughput.

V. EXPERIMENTAL RESULTS

A. Experimental Setup

Models and datasets: We use three different types of diffusion models: Stable Diffusion v1.5 [17] (text-conditioned), Diffusion Transformer (DiT) [18] as a class-conditioned model, and DDIM [19] that is an unconditional model. The datasets are MS COCO [20], ImageNet [21], and CelebA-HQ (loaded from Hugging Face[2]), respectively. The resolution of the images in the first model is 512×512 and it is 256×256 for the other two models. All experimental settings (e.g., variance schedule, guidance scale, scheduler) follow the official implementation, with 50 time steps. We also use LDM [17] and IDDPM [22] for comparison with related work.

Quantization baselines: We compare the performance of our approach against these state-of-the-art: Q-Diffusion [8], PTQ4DM [9], and PTQD [10].

Implementation: We implement DM-Tune using PyTorch with model-agnostic quantization achieved via hook functions registered conditionally to specific layers, timesteps, and tensor portions. During the forward pass, activations are quantized and de-quantized, while the backward pass employs straight-through estimation (STE) [23]. Batch normalizations and activation functions remain in high precision. Models run on an A100 GPU (40 GB) for noise-tuning and quality evaluation, and on an L4 GPU (24 GB) for runtime assessment due to FP8 support. Diffusion models are evaluated with DGM-Eval [14].

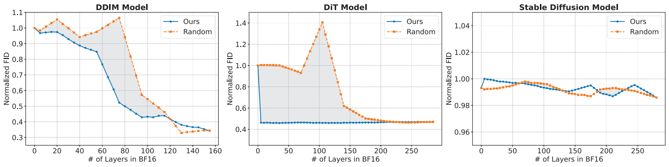

Fig. 5. Comparison of FID between our sensitivity-based quantization strategy and random precision selection, gradually shifting from entirely low-precision to entirely high-precision in the layers.

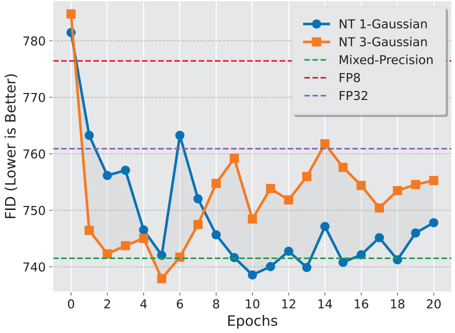

Fig. 6. Comparison of FID scores across epochs for two noise-tuning methods with one and three Gaussian terms for Stable Diffusion model using MS COCO evaluation dataset.

Noise-tuning uses the Adam optimizer with a learning rate of *1e-3*, 4K training samples, 1K evaluation samples, and batch sizes of 8 (Stable Diffusion), 64 (DiT), and 32 (DDIM), halved during noise-tuning to avoid out-of-memory (OoM) errors.

B. Mixed-Precision Sensitivity

We first demonstrate the effectiveness of our mixed-precision sensitivity analysis before applying prompt-aware and timestep-aware quantization. Sensitivity criteria are evaluated against a random baseline, which assigns high- and low-precision layers randomly based on a fixed split. Fig. 5 compares FID scores for three models. For DiT, our method effectively identifies sensitive layers, achieving a larger performance gap over the random approach. The gap is smaller for DDIM, while for Stable Diffusion, layer precision choices have minimal impact on accuracy.

C. Training

We evaluate noise-tuning performance across training epochs, comparing it to the FP-MP ground truth, FP8, and FP32 to assess convergence. To demonstrate its superior expressivity, we also compare noise-tuning with one Gaussian term (*NT 1_Gaussian*) to three terms (*NT 3_Gaussian*). Fig. 6 presents FID metrics during training on the MS COCO dataset

[2] https://huggingface.co/google/ddpm-celebahq-256

TABLE II

COMPARING DM-TUNE PERFORMANCE WITH STATE-OF-THE-ART. NT IS NOISE-TUNING.

Model/Dataset	Method	FID ↓	Precision ↑	Recall ↑	Density ↑	Coverage ↑	KD ↓	SW-approx ↓
IDDPM (ImageNet 64×64)	FP32 (Baseline)	282.02	0.80	0.84	0.77	0.81	0.15	0.20
	FP8	519.50	0.73	0.56	0.68	0.48	0.66	0.36
	INT8	4186.98	0.0	0.0	0.0	0.0	31.23	1.84
	PTQ4DM	376.09	0.78	0.80	0.78	0.72	0.29	0.27
	Ours (w/o NT)	**256.17**	**0.82**	**0.87**	0.74	**0.84**	**0.14**	**0.16**
LDM (ImageNet 64×64)	FP32 (Baseline)	265.83	0.98	0.42	0.98	0.89	0.059	0.12
	FP8	346.77	0.95	0.41	0.99	0.87	0.28	0.26
	INT8	3519.99	0.45	0.0	0.11	0.0	19.35	1.52
	PTQD	226.51	0.98	0.53	0.97	0.91	0.041	0.10
	Ours (w/o NT)	**232.79**	0.96	**0.51**	0.98	**0.93**	**0.048**	**0.09**

TABLE III

COMPARISON OF NOISE-TUNING OPTIMIZATION TECHNIQUES FOR MATRIX MULTIPLICATION (1K×1K MATRICES). NORMALIZED RUNTIME IS SHOWN.

FP8	NT (8 bit) w/o opt	NT (8 bit) O1	NT (8 bit) O1 + O2	NT (8 bit) O1 + O2 + O3
1.00X	1.12X	1.04X	1.02X	1.01X

with the Stable Diffusion model. *NT 3_Gaussian* converges faster and even surpasses the FP-MP ground truth. Notably, FP-MP performance is achieved in just 5-10 epochs, underscoring the efficiency and low training cost of our approach.

D. Image Generation Performance

We evaluate DM-Tune's performance against other quantization methods. Table I compares results for three diffusion models using 1K samples and DGM-Eval metrics. For Stable Diffusion and DiT, DM-Tune without noise-tuning (NT) outperforms full precision, with noise-tuning further improving results. For the unconditional DDIM model, DM-Tune surpasses the FP32 baseline only with noise-tuning, likely due to initial mixed-precision noise and subsequent fine-tuning with nonlinear parameters. Overall, DM-Tune improves quality, diversity, prompt alignment, and originality compared to full precision, especially for conditional models. It outperforms Q-Diffusion in most cases, as their integer-based quantization limits dynamic range and lacks techniques to exceed full-precision quality. Fig. 1(a) highlights I-MP's challenges with high-fidelity image generation for human faces.

We use the DINOv2-ViT model as the feature extractor to ensure fairness in evaluation, leading to higher FID results than those in related work. To address this, we re-evaluate related work using our method and feature extractor. Since prior methods support limited models and datasets due to their calibration techniques, we compare DM-Tune with them on the datasets they support. Table II shows a comparison using 5K samples with DGM-Eval metrics. DM-Tune outperforms prior methods in most scenarios, though PTQD achieves better results on some metrics.

E. Runtime

We evaluate the runtime of the proposed noise-tuning, starting with a single matrix multiplication kernel with a NT head. Table III compares the runtime of various optimizations for a kernel fused with noise-tuning (three Gaussian terms) on 1k×1k matrices, averaged over 10 runs and normalized to FP8 without noise-tuning. With all optimizations, the runtime matches FP8, highlighting the efficiency of our approach. Fig. 7 compares the model runtime and performance against FP32, FP8, and state-of-the-art methods. Fully optimized noise-tuning adds minimal overhead to FP8 while delivering a 5.2× average performance boost over prior art.

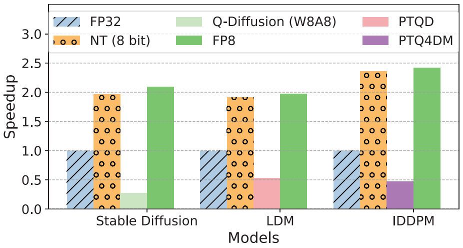

Fig. 7. Normalized model performance comparison of noise-tuning against state-of-the-art methods.

VI. RELATED WORK

Diffusion Model Quantization: Q-Diffusion [8] introduces post-training quantization (PTQ) using an 8/4-bit integer format with a timestep-aware calibration sampling mechanism. Similarly, PTQ4DM [9] applies 8-bit integer quantization but focuses on smaller models like DDPM and DDIM. To our knowledge, no prior work has demonstrated that quantization enhances image generation in diffusion models.

Noise Modeling: [10] introduces a mixed-precision integer quantization method that models quantization noise and applies corrections to reconstruct full-precision outputs. Unlike our approach, which uses mixed precision as ground truth, they assume full precision yields the best performance. Their reliance on statistical and manual profiling limits applicability to specific data formats and models.

VII. CONCLUSION

In this work, we introduced DM-Tune, a framework combining unified low-precision quantization with noise-tuning to enhance diffusion model efficiency and image quality. DM-Tune leverages inherent model noise to surpass the limitations of traditional mixed-precision methods, improving generation metrics. Experiments show that DM-Tune matches or outperforms full-precision models in quality, diversity, and text-to-image alignment while reducing inference time. Optimized GPU kernels further accelerate deployment. Future work will explore extending DM-Tune to lower-precision formats (FP4/6) and quantizing additional components like encoder-decoder structures to enhance runtime.

ACKNOWLEDGEMENTS

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, ComPort: Rigorous Testing Methods to Safeguard Software Porting, under Award Number 78284. The Pacific Northwest National Laboratory is operated by Battelle for the U.S. Department of Energy under Contract DE-AC05-76RL01830. This work was also supported in part by NSF under Awards SHF-2326494.

References

[1] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," 2021. [Online]. Available: https://arxiv.org/abs/2102.12092

[2] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi, "Photorealistic text-to-image diffusion models with deep language understanding," 2022. [Online]. Available: https://arxiv.org/abs/2205.11487

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2022. [Online]. Available: https://arxiv.org/abs/2112.10752

[4] Y. Liu, K. Zhang, Y. Li, Z. Yan, C. Gao, R. Chen, Z. Yuan, Y. Huang, H. Sun, J. Gao, L. He, and L. Sun, "Sora: A review on background, technology, limitations, and opportunities of large vision models," 2024. [Online]. Available: https://arxiv.org/abs/2402.17177

[5] C. Liu, C. Wu, S. Cao, M. Chen, J. C. Liang, A. Li, M. Huang, C. Ren, Y. N. Wu, D. Liu, and T. Geng, "Diff-pic: Revolutionizing particle-in-cell nuclear fusion simulation with diffusion models," in *The Thirteenth International Conference on Learning Representations*, 2025.

[6] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, and R. Rombach, "Sdxl: Improving latent diffusion models for high-resolution image synthesis," 2023. [Online]. Available: https://arxiv.org/abs/2307.01952

[7] C. Liu, R. Song, C. Wu, P. Haghi, and T. Geng, "Instatrain: Adaptive training via ultra-fast natural annealing within dynamical systems," in *The Thirteenth International Conference on Learning Representations*, 2025.

[8] X. Li, Y. Liu, L. Lian, H. Yang, Z. Dong, D. Kang, S. Zhang, and K. Keutzer, "Q-diffusion: Quantizing diffusion models," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 17 489–17 499.

[9] Y. Shang, Z. Yuan, B. Xie, B. Wu, and Y. Yan, "Post-training quantization on diffusion models," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 1972–1981.

[10] Y. He, L. Liu, J. Liu, W. Wu, H. Zhou, and B. Zhuang, "Ptqd: accurate post-training quantization for diffusion models," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NeurIPS '23. Red Hook, NY, USA: Curran Associates Inc., 2024.

[11] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015. [Online]. Available: https://arxiv.org/abs/1505.04597

[12] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," 2020. [Online]. Available: https://arxiv.org/abs/2006.11239

[13] A. M. Alaa, B. van Breugel, E. Saveliev, and M. van der Schaar, "How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models," 2022. [Online]. Available: https://arxiv.org/abs/2102.08921

[14] G. Stein, J. Cresswell, R. Hosseinzadeh, Y. Sui, B. Ross, V. Villecroze, Z. Liu, A. L. Caterini, E. Taylor, and G. Loaiza-Ganem, "Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models," in *Advances in Neural Information Processing Systems*, vol. 36, 2023.

[15] M. F. Naeem, S. J. Oh, Y. Uh, Y. Choi, and J. Yoo, "Reliable fidelity and diversity metrics for generative models," 2020. [Online]. Available: https://arxiv.org/abs/2002.09797

[16] J. Ho and T. Salimans, "Classifier-free diffusion guidance," 2022. [Online]. Available: https://arxiv.org/abs/2207.12598

[17] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," 2022. [Online]. Available: https://arxiv.org/abs/2112.10752

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

[19] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=St1giarCHLP

[20] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015. [Online]. Available: https://arxiv.org/abs/1405.0312

[21] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[22] A. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," 2021. [Online]. Available: https://arxiv.org/abs/2102.09672

[23] P. Yin, J. Lyu, S. Zhang, S. Osher, Y. Qi, and J. Xin, "Understanding Straight-Through Estimator in Training Activation Quantized Neural Nets," 2019.