FP4DiT: Towards Effective Floating Point Quantization for Diffusion Transformers

Ruichen Chen*   Keith G. Mills*   Di Niu
Department of Electrical and Computer Engineering
University of Alberta
{ruichen1, kgmills, dniu}@ualberta.ca

Abstract

Diffusion Models (DM) have revolutionized the text-to-image visual generation process. However, the large computational cost and model footprint of DMs hinders practical deployment, especially on edge devices. Post-training quantization (PTQ) is a lightweight method to alleviate these burdens without the need for training or fine-tuning. While recent DM PTQ methods achieve W4A8 on integer-based PTQ, two key limitations remain: First, while most existing DM PTQ methods evaluate on classical DMs like Stable Diffusion XL, 1.5 or earlier, which use convolutional U-Nets, newer Diffusion Transformer (DiT) models like the PixArt series, Hunyuan and others adopt fundamentally different transformer backbones to achieve superior image synthesis. Second, integer (INT) quantization is prevailing in DM PTQ but doesn't align well with the network weight and activation distribution, while Floating-Point Quantization (FPQ) is still under-investigated, yet it holds the potential to better align the weight and activation distributions in low-bit settings for DiT. In response, we introduce FP4DiT, a PTQ method that leverages FPQ to achieve W4A6 quantization. Specifically, we extend and generalize the Adaptive Rounding PTQ technique to adequately calibrate weight quantization for FPQ and demonstrate that DiT activations depend on input patch data, necessitating robust online activation quantization techniques. Experimental results demonstrate that FP4DiT outperforms integer-based PTQ at W4A6 and W4A8 precision and generates convincing visual content on PixArt-α, PixArt-Σ and Hunyuan in terms of several T2I metrics such as HPSv2 and CLIP. Code is available at https://github.com/cccrrrccc/FP4DiT.

1. Introduction

Diffusion Transformers (DiT) [41] are on the forefront of open-source generative visual synthesis. In contrast to earlier text-to-image (T2I) Diffusion Models (DMs) like Stable Diffusion v1.5 [44] and Stable Diffusion XL [42] that utilize a classical U-Net structure, DiTs such as PixArt-α [3], PixArt-Σ [2] and Stable Diffusion 3 (SD3) [7] leverage streamlined, patch-based Transformer architectures to generate high-resolution images.

Nevertheless, similar to U-Nets, DiTs utilize a lengthy denoising process that incurs a high computational inference cost. One method to alleviate this burden is quantization [5, 57], which reduces the bit-precision of neural network weights and activations. As the first Post-Training Quantization (PTQ) schemes for DMs, PTQ4DM [45] and Q-Diffusion [23] demonstrate that the range and distribution of U-Net activations crucially depend on the diffusion timestep. More recent state-of-the-art works likeTFMQ-DM [14] specialize quantization for U-Net timestep conditioning which may not generalize to newer DiTs. Further, methods like ViDiT-Q [58] adapt outlier suppression technique [55] to DiTs, but overlook broader advantages of prior DM PTQ like weight reconstruction [24, 34].

Moreover, the prevailing datatypes in existing DM PTQ literature [8, 14, 23, 33, 45, 47, 58] are integer-based (INT), which provide uniformly distributed values [36] unlike the non-uniform distribution of weights and activations in modern neural networks [46]. Thus, PTQ for text-to-image (T2I) DiTs below W4A8 precision (4-bit weights and 8-bit activations) without severely compromising generation quality remains an open challenge.

In this paper, we present FP4DiT, which achieves W4A6 PTQ on Diffusion Transformers with non-uniform Floating-Point Quantization (FPQ) [20], thus achieving high quantitative and qualitative T2I performance. Besides, by introducing FPQ, FP4DiT not only aligns the quantization levels better with the weight and activation distribution with negligible computational overhead, it also massively reduces the cost of weight calibration by over 8×. Our detailed contributions are summarized as follows:
1. We apply FPQ to DiT to address the misalignment between the existing DM PTQ literature and the non-

---

*Equal contribution.

uniform distribution of network weights and activations.

2. We reveal the critical role of preserving the sensitive interval of DiT's GELU activation function and propose a mixed-format FPQ method tailored for DiT.

3. We examine the adaptive rounding (AdaRound) [34] mechanism, originally designed for integer PTQ, and reveal a performance-hampering design limitation when applied to FPQ. In response, we introduce a novel mathematical scaling mechanism that greatly improves the performance of AdaRound when utilized in the FPQ scenario.

4. We analyze DiT activation distributions and visualize how they contrast to those of convolutional U-Nets, especially with respect to diffusion timesteps. Specifically, while U-Net activation ranges *shrink* with timestep progression, DiT activations ranges instead *shift* over time. To address this, we implement an effective online activation quantization [53, 57] scheme to accommodate DiT activations.

We apply FP4DiT as a PTQ method on T2I DiT models, namely PixArt-α, PixArt-Σ, and Hunyuan. To verify the effectiveness of FP4DiT, we conduct extensive experiments on T2I tasks such as the Human Preference Score v2 (HPSv2) benchmark [52] and MS-COCO dataset [27], to outperform existing methods like Q-Diffusion [23], TFMQ-DM [14] and ViDiT-Q [58] at the W4A8 and W4A6 precision levels. Additionally, we perform a human preference study which demonstrates the superiority of FP4DiT-generated images.

2. Related Work

Diffusion Transformers (DiT) [41] replace the classical convolutional U-Net [44] backbone with a modified Vision Transformer (ViT) [6] to increase scalability. Although the introduction of DiT architectures in newer DMs [2, 3, 7, 21, 25, 56] enables the generation of high-quality visual content [1], DiTs still suffer from a computationally expensive diffusion process, rendering deployment on edge devices impractical and cumbersome. Further, addressing this weakness for DiTs poses unique challenges compared to U-Nets, and is a focus of this work.

Quantization is a neural network compression technique that involves reducing the bit-precision of weights and activations to lower hardware metrics like model size, inference latency and memory consumption [35]. The objective of quantization research is to reduce bit-precision as much as possible while preserving overall model performance [31]. There are two main classes of quantization: Quantization-Aware-Training (QAT) [8, 10, 48] and Post-Training Quantization (PTQ) [24, 33]. Specifically, PTQ is more lightweight and neither requires re-training nor substantial amounts of data. Rather, PTQ requires a small amount of data to calibrate quantization scales [34],

typically in a block-wise manner [24]. However, while most PTQ methods rely on uniformly-distributed integer (INT) quantization techniques [16, 19], recent literature highlights the advantages of low-bit floating point quantization (FPQ) [28, 53] for LLMs. Therefore, in this paper we investigate the challenges in applying FPQ to DM PTQ.

The denoising process of DMs brings new challenges for PTQ compared with traditional neural networks. The earliest DM PTQ research [45] reveals the significant activation range changes across different denoising timesteps. Q-Diffusion [23] samples calibration data across different denoising timesteps to address this challenge. TDQ [47] calibrates an individual set of the quantization parameters across different time steps, offering a more fine-grained approach to managing temporal dependencies. TFMQ-DM [15] highlights the sensitivity of temporal features in U-Nets and introduce a calibration objective aimed at better preserving temporal characteristics. However, the above works are specific to U-Net architectures while DiT architectures feature distinct activation characteristics. Further, although some early research on FPQ for DiT models exists [29] exists, such approaches are limited by the increased online inference cost and have yet to be rigorously evaluated on T2I DiT models like the PixArt models and Hunyuan-DiT [25]. ViDiT-Q [58] utilizes fine-grained techniques including channel balancing, mixed-precision and outlier suppression from LLM literature [55] to quantize DiTs, it does not incorporate weight reconstruction [24, 34] as in existing DM PTQ methods, thus limit the model's ability to maintain model performance when utilizing low bit-width quantization. In contrast, this work aggregates the cumulative knowledge of existing DM PTQ methods and refines them for application on T2I DiT models.

3. Methodology

In this section, we present our PTQ solution for the T2I DiT model. First, we analyze the sensitivity of the DiT block in the PixArt and Hunyuan model and propose a mixed FP format for the FP4 weight quantization. Second, we propose a scale-aware AdaRound tailored for FP weight quantization. Lastly, we investigate and contrast U-Net and DiT activation distribution information.

3.1. Uniform vs. Non-Uniform Quantization

Quantization compresses neural network size by reducing the bit-precision of weights and activations, e.g. rounding from 32/16-bit datatypes into an n-bit quantized datatype, where $n \leq 8$ typically. For instance, we can perform uniform integer (INT) quantization on a tensor $\mathbf{X}$ to round it into a lower-bit representation $\mathbf{X}^{(\text{int})}$ as follows:

$$\mathbf{X}^{(\text{int})} = \text{clip}\left(\left\lfloor \frac{\mathbf{X}}{s} \right\rceil + z, x_{\min}, x_{\max}\right) \qquad (1)$$
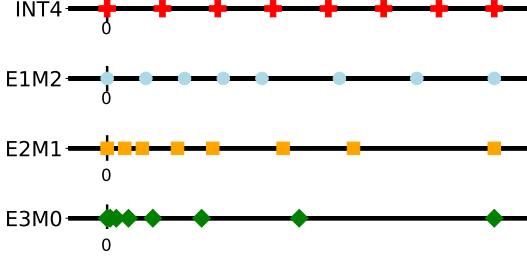
Figure 1. Value distributions for INT4 and three variants of FP4: E1M2, E2M1 and E3M0. Note that E0M3 is INT4. Observe how INT4 values are evenly distributed, while FP4 values cluster closer to the origin as the number of exponent (E) bits increases.

where s is scale, z is the zero point, and $\lfloor \cdot \rceil$ is operation rounding-to-nearest. INT quantization rounds the values to a range with 2^n points. Specifically, the range is always a uniform grid, whose size decreases by half each time n decreases by 1.

In contrast, Floating-Point Quantization (FPQ) uses standard floating-point numbers as follows:

$$f = (-1)^{d_s} 2^{p-b} \left(1 + \frac{d_1}{2} + \frac{d_2}{2^2} + \cdots + \frac{d_m}{2^m}\right) \quad (2)$$

where $d_s \in \{0, 1\}$ is the sign bit and b is the bias. $p = d_1 + d_2 * 2 + \cdots + d_e * 2^{e-1}$ represents the e-bit exponent part while $\left(1 + \frac{d_1}{2} + \frac{d_2}{2^2} + \cdots + \frac{d_m}{2^m}\right)$ represents the m-bit mantissa part. Note that $d_i \in \{0, 1\}$ for bits in both the mantissa and the exponent part. The FP format can be seen as multiple consecutive m-bit uniform grids with different exponential scales. Therefore, the FPQ is operated similarly to Equation 1, with distinct scaling factors applied to values across varying magnitudes.

The key advantage of FPQ, especially at low-bit precision for quantization, is that they enjoy a richer granularity of value distributions owing to the numerous ways we can vary the allocation of exponent and mantissa bits. This is analogous to the introduction of the 'BFloat16' [18] format, which achieves superiority over the older IEEE standard 754 'Float16' [17] in certain deep machine algorithms [22] by allocating 8-bits towards the exponent, as the larger 'Float32' format does. Broadly, an n-bit floating point datatype posses $n - 1$ possible distributions as $m \in [0, n - 1]$, and even adopts the uniform distribution of the corresponding n-bit integer format when $m = n - 1$.

Figure 1 visualizes this advantage by showing the discrete value distribution of INT4 and FP4 under different FP formats. The bits allocation between the mantissa and exponent significantly influences the performance of quantization as depicted. While the flexibility of floating-point format benefits the quantization, improper FP format can result in sub-optimal performance [46]. Hence, in the following section, we present our analysis of the DiT blocks and introduce our method, which adjusts the FP format when
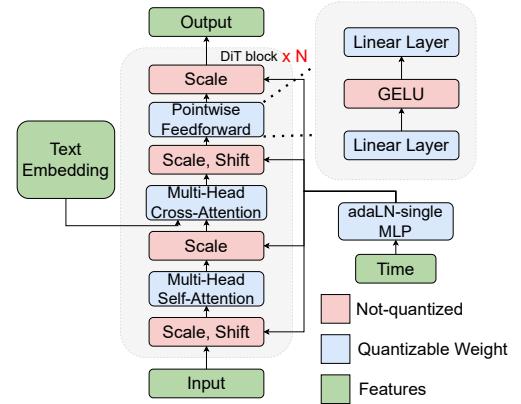

Figure 2. T2I DiT block diagram. In PixArt-α/Σ, all DiT blocks share the same adaLN-single MLP for time conditions. The scale and shift for layer normalization in DiT blocks depend on the embedding from adaLN-single and the layer-specific training embedding. Colored blocks distinguish the weight layers we can quantize from activation and normalization functions.
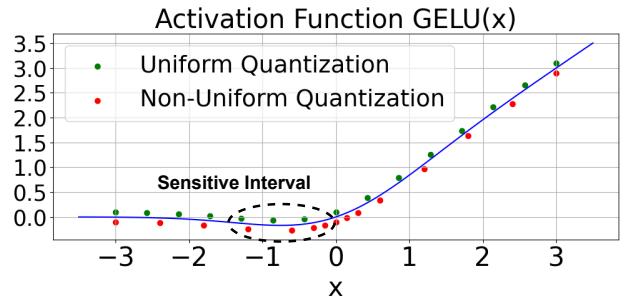

Figure 3. The GELU activation and its sensitive interval. With the same amount of discrete values, non-uniform quantization can better capture the sensitive interval.

quantizing different DiT weights.

3.1.1. Optimized FP Formats in DiT Blocks.

Figure 2 illustrates the structure of a T2I DiT Block. In a DiT block, the Pointwise Feedforward is unique in that it consists of a non-linear GELU activation flanked by linear layer before and after. GELU, plotted in Figure 3 contains a sensitive region where the function returns a negative output. Interestingly, Reggiani et al., 2023 [43] show that focusing on this sensitive interval helps reduce the mean-squared error when approximating GELU using Look-Up Tables (LUTs) or breakpoints. Building on this insight, we apply denser floating point formats, e.g., E3M0, to the first pointwise linear layer. This allocates more values closer to zero, i.e., where the GELU sensitive interval lies, thereby enhancing the precision of the approximation. Due to space constraints, we provide further information and experiment
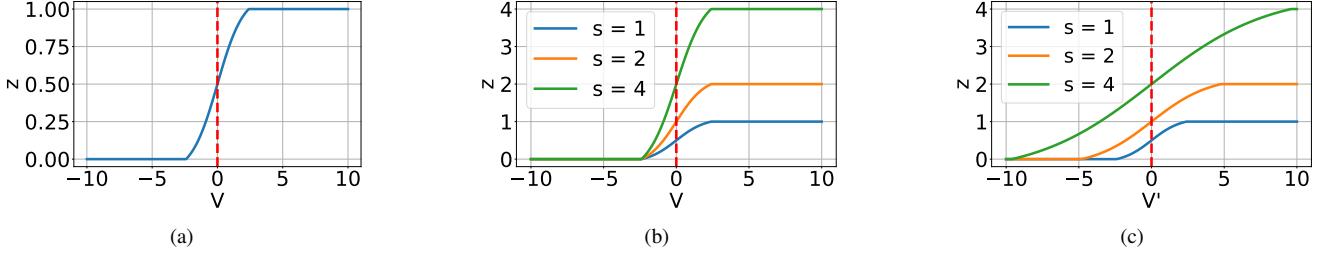
3

Figure 4. (a) The binary gate function of INT AdaRound. All gates are identical because there is only one scale in INT quantization. (b) The binary gate functions of origin AdaRound on FP quantization. (c) The binary gate functions of scale-aware AdaRound. The red dashed line indicates the demarcation of rounding up (right) or down (left). Our scale-aware AdaRound normalizes the slope near the turning point, which stabilizes the optimization and helps improve the quantization performance.

evidence in the Supplementary Materials.

3.2. AdaRound for FP

By default, quantization is rounding-to-nearest, e.g., Eq. 1. AdaRound [34] show that rounding-to-nearest is not always optimal and instead apply second-order Taylor Expansion on the loss degradation from weight perturbation $\Delta\mathbf{w}$ caused by quantization:

$$E[\Delta L(\mathbf{w})] \approx \Delta\mathbf{w}^T\mathbf{g}^{(\mathbf{w})} + \frac{1}{2}\Delta\mathbf{w}^T\mathbf{H}^{(\mathbf{w})}\Delta\mathbf{w}. \quad (3)$$

The gradient term $\mathbf{g}^{(\mathbf{w})}$ is close to 0 as neural networks are trained to be converged. Hence, the loss degradation is determined by the Hessian matrix $\mathbf{H}^{(\mathbf{w})}$, which defines the interactions between different perturbed weights in terms of their joint impact on the task loss. The rounding-to-nearest is sub-optimal because it only considers the on-diagonal elements of $\mathbf{H}^{(\mathbf{w})}$. However, optimizing via a full Hessian matrix is infeasible because of its computational and memory complexity issues. To tackle these issues, the authors make assumptions such as each non-zero block in $\mathbf{H}^{(\mathbf{w})}$ corresponds to one layer, and then propose an objective function:

$$\arg\min_{\mathbf{V}} \|W\mathbf{x} - \widetilde{W}\mathbf{x}\|_F^2 + \lambda f_{\text{reg}}(\mathbf{V}), \quad (4)$$

The optimization objective is to minimize the Frobenius norm of the difference between the full-precision output $W\mathbf{x}$ and the quantized output $\widetilde{W}\mathbf{x}$ for each layer and $f_{\text{reg}}(\mathbf{V})$ is a differentiable regularizer to encourage the variable $\mathbf{V}$ to converge. BRECQ [24] proposed a similar block-wise optimization objective that further advances the performance of weight reconstruction in PTQ.

In detail, $\widetilde{W}$ is defined as follows:

$$\widetilde{W} = s \cdot \text{clip}\left(\left\lfloor \frac{W}{s} \right\rfloor + h(\mathbf{V}), min, max\right) \quad (5)$$

where min and max denotes the quantization threshold. $h(\mathbf{V})$ is the rectified sigmoid function proposed by [30]:

$$h(\mathbf{V}) = \text{clip}\left(\sigma(\mathbf{V})(\zeta - \gamma) + \gamma, 0, 1\right) \quad (6)$$

where $\sigma(\cdot)$ is the sigmoid function and ζ and γ are fixed to 1.1 and -0.1. During optimization the value of $h(\mathbf{V})$ is continuous, while during inference its value will be set to 0 or 1 indicating rounding up or down.

3.2.1. Scale-aware AdaRound.

AdaRound has been widely adopted to improve the performance of quantized neural networks [23, 24] in low-bit settings like 4-bit weights. However, AdaRound assumes weight quantization to low-bit integer formats, like INT4 rather than low-bit FP formats [49], where non-uniform value distribution (Fig. 1) may introduce unique challenges.

Specifically, we identify that the original INT-based AdaRound assumes the scale s is consistent across different quantized values. However, this does not hold for FPQ, where there are 2^E scales. Therefore, we propose scale-aware AdaRound which improves the performance and leads to faster convergence.

Our scale-aware AdaRound inherits Equation 4 as the learning objective because reducing the layer-wise and block-wise quantization error is the common goal of FPQ and INT quantization. Differently, We modified the $\widetilde{W}$ as:

$$\widetilde{W} = s \cdot \text{clip}\left(\left\lfloor \frac{W}{s} \right\rfloor + h'(\mathbf{V}'), min, max\right) \quad (7)$$

$$h'(\mathbf{V}') = \text{clip}\left(\sigma(\frac{\mathbf{V}'}{s})(\zeta - \gamma) + \gamma, 0, 1\right) \quad (8)$$

where $h'(\cdot)$ is the scale-aware rectified sigmoid function and V' is a new continuous variable we optimized over.

The rectified sigmoid function functions as binary gates that control the rounding of weights. Specifically, the gate function $z = s \cdot h(V)$ is optimized according to Equation 4. Figure 4a shows those binary gates in INT AdaRound. The gates are equivalent across all the weights, which matches the even distribution of INT quantization. In Figure 4b, we show the origin AdaRound's binary gates under different scales. The gates' slope depends on their scale, which causes imbalanced update during the gradient descent. In
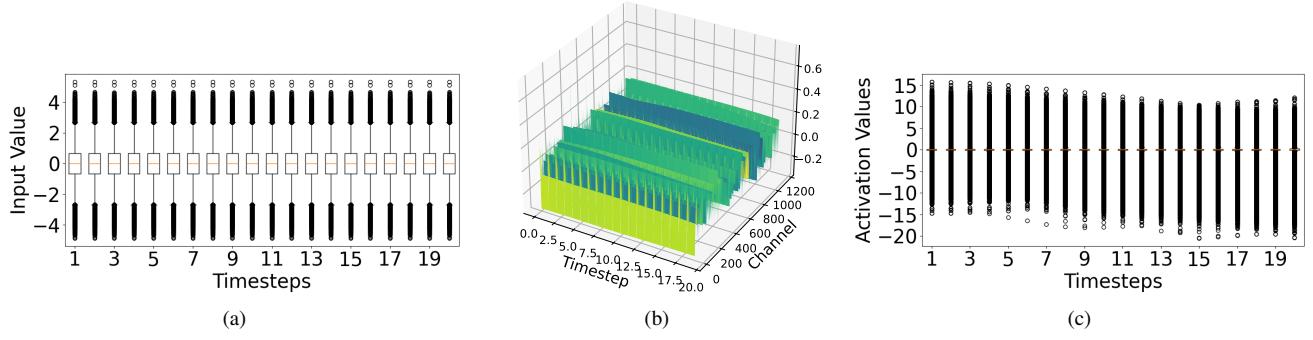
4

Figure 5. (a) Different timestep input values for PixArt-α on 128 images sampled from MS-COCO. The input does not shrink progressively across timesteps like U-Net DM. (b) The time-embedded scale for the output of the 7th DiT block's FeedForward. It is almost constant across timesteps. (c) The output of the 7th DiT block. Its range tends to remain constant but shifts as a function of time.
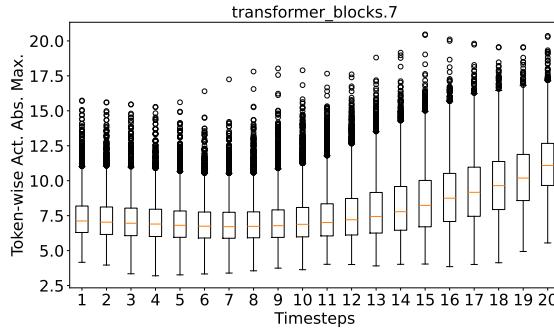

Figure 6. The distribution of the absolute maximum for each token's activation among 4096 tokens in the PixArt-α model. The distribution demonstrates a strong patch dependency in the DiT activation.

Figure 4c, we show the binary gates of our scale-aware AdaRound. In contrast, the gates' slope is normalized to the same level, which makes the weight reconstruction much more stable and thus aids the quantization. For the mathematical proof of our scale-aware AdaRound, please refer to the Supplementary Materials.

3.3. Token-wise Activation Quantization

Prior DM PTQ approaches [11, 15] use a calibration dataset to learn temporally-aware [10] activation quantization scales. This approach is predicated on knowledge of how U-Net activation distributions change as a function of denoising timesteps, i.e., activation ranges taper-off towards the end of the denoising process [23, 45].

In contrast, we show that this assumption does not hold for DiT models. First, we collect input activations of PixArt-α across 20 timesteps, revealing that the activation range remains stable over time, as shown in Figure 5a. We then analyze the Adaptive Layer Normalization (AdaLN) in the PixArt DiT model in Figure 2. Since the feed-forward scale directly influences the DiT block output range, we vi-

Method	Precision	CLIP ↑
No Quantization	W16A16	0.3075
Temporally-Aware Act. Quant.	W4A6	0.2012
Token-wise Act. Quant.	W4A6	0.3036
Temporally-Aware Act. Calib.	W4A8	0.2410
Token-wise Act. Quant.	W4A8	0.3120

Table 1. CLIP score [12] reported when quantizing PixArt-α to 4-bit weights (W4) and 8 or 6-bit activations (A8 and A6) using the online token-wise method and temporarally-aware scale calibration. Specifically, we generate 1k images per configuration using COCO [26] prompts and compare against the validation set. Higher CLIP means better prompt-image alignment.

sualize the feed-forward scale in the first layer in Figure 5b and the output of the 7th DiT block in Figure 5c. These figures demonstrate that the value of activations is primarily controlled by channels opposed to timesteps, and that the width of the activation range tends to remain constant, but shifts as a function of time.

Further, we plot the token-wise activation range in Figure 6. This plot visualizes the absolute maximum activation of each image patch (token) across time. The results indicate that the activation range varies significantly, even among tokens within the same timestep.

Recent works by Microsoft [53, 57] suggest that token-wise online activation quantization approaches may yield superior results when quantizing transformer activations, especially when paired with kernel fusion [51] techniques. Table 1 applies this hypothesis to the DiT scenario. Specifically, we apply simple min-max quantization to reduce weight precision of PixArt-α to 4-bits (W4), then consider 8-bit (A8) and 6-bit (A6) activation quantization. In both scenarios, we observe CLIP performance that is closer to the full precision model using token-wise activation quantization as opposed to the traditional, temporally-aware scale calibration technique which is designed for U-Nets.

As such, we consider token-wise activation quantization throughout the remainder of this work by substituting it into U-Net baselines like Q-Diffusion [23] and TFMQ-DM [15].

4. Results

In this section we conduct experiments to verify the efficacy of FP4DiT. We elaborate on our experimental setup and then compare FP4DiT to several baselines approaches to highlight its competitiveness in terms of quantitative metrics and qualitative image generation output. Specifically, we consider three text-to-image (T2I) models: PixArt-α [3], PixArt-Σ [2] and Hunyuan [25]. We also conduct several ablation studies to verify the components of our method. Finally, we report several hardware metrics tabulating the cost savings and throughput of FP4DiT.

4.1. Experimental Settings

We use the HuggingFace Diffusers library [50] to instantiate the base DiT models in W16A16 bit-precision and consider the default values for inference parameters like number of denoising steps and classifier-free guidance (CFG) scale. We quantize weights to 4-bit precision FP format. Specifically, we set the weight format for the first linear layer in each pointwise feed-forward to be E3M0. We quantize all other weights to E2M1 for PixArt-α and Hunyuan, and E1M2 for PixArt-Σ. Further details on this decision are provided in the Supplementary Materials. Finally, our weight quantization is group-wise [9, 40] along the output channel dimension with a group size of 128.

We perform weight quantization calibration using our scale-aware AdaRound and BRECQ. Weight calibration requires a small amount of calibration data. We use 128 (64 for Hunyuan) image-text pairs from the MS-COCO 2017 train [26] dataset and calibrate for 2.5k iterations per DiT block or layer. Next, we perform activation quantization to 8 or 6-bit precision using min-max token-wise quantization from ZeroQuant [53, 57]. We provide further hyperparameter details in the Supplementary Materials.

4.2. Main Results

In our experiment, we consider three baseline approaches: Q-Diffusion [23], TFMQ-DM [15] and ViDiT-Q [58]. Note that while Q-Diffusion and TFMQ-DM are originally designed for U-Nets, we modify these approaches to use the same online token-wise activation quantization as FP4DiT per Table 1, while ViDiT-Q uses this mechanism by default. Further, note that ViDiT-Q uses mix-precision meaning some of their layers are not quantized to 4-bits. For the sake of convenience, we use their mix-precision as a W4 baseline to conduct our experiments.

We generate 512×512 resolution images using PixArt-α and 1024×1024 for PixArt-Σ and Hunyuan. For eval-

Model	Method	Precision	HPSv2 ↑	CLIP ↑
PixArt-α	Full Precision	W16A16	31.01	0.3075
	Q-Diffusion	W4A8	23.17	0.3017
	TFMQ-DM	W4A8	*25.99*	*0.3066*
	ViDiT-Q	W4A8	17.42	0.2900
	FP4DiT (ours)	W4A8	**27.43**	**0.3076**
	Q-Diffusion	W4A6	12.5	0.2868
	TFMQ-DM	W4A6	*23.29*	*0.3015*
	ViDiT-Q	W4A6	16.44	0.2827
	FP4DiT (ours)	W4A6	**23.55**	**0.3031**
PixArt-Σ	Full Precision	W16A16	31.67	0.3139
	Q-Diffusion	W4A8	*26.13*	*0.3050*
	TFMQ-DM	W4A8	23.62	0.2975
	ViDiT-Q	W4A8	26.07	0.2562
	FP4DiT (ours)	W4A8	**26.27**	**0.3064**
	Q-Diffusion	W4A6	22.67	*0.3027*
	TFMQ-DM	W4A6	18.53	0.2346
	ViDiT-Q	W4A6	*23.37*	0.2425
	FP4DiT (ours)	W4A6	**25.40**	**0.3040**
Hunyuan	Full Precision	W16A16	32.08	0.3102
	Q-Diffusion	W4A8	24.96	0.3006
	TFMQ-DM	W4A8	*27.56*	*0.3075*
	FP4DiT (ours)	W4A8	**27.78**	**0.3102**
	Q-Diffusion	W4A6	*14.83*	0.2277
	TFMQ-DM	W4A6	13.68	*0.2520*
	FP4DiT (ours)	W4A6	**15.00**	**0.2562**

Table 2. Quantitative evaluation results for PixArt-α, PixArt-Σ and Hunyuan in terms of average score on the HPSv2 benchmark and average CLIP score using 10k (3k for Hunyuan) COCO 2017 prompt-image pairs. Best and second best results in **bold** and *italics*, respectively.

uation, we primarily consider the Human Preference Score v2 (HPSv2) [52] benchmark. Specifically, HPSv2 considers four image categories: animation, concept-art, painting and photography, and estimates the human preference score of an image generated using a prompt with respect to one category. Each category contains 800 prompts, requiring 3.2k images be generated to fully evaluate. The final HPSv2 score is the average estimated human preference across all four categories. Additionally, we measure the CLIP [12] score using prompts from the MS-COCO 2017 [26] validation set.

Table 2 compares our method with the stated baseline approaches at the W4A8 and W4A6 precision levels. FP4DiT consistently outperforms all other methods across three different base models at every precision level. Specifically, for the PixArt models, W4A6 precision causes the baselines to experience a significant performance drop, while FP4DiT still maintains a balanced trade-off between accuracy and efficiency. Also, results on Hunyuan further demonstrate the efficacy of our method at A8 precision, and although A6 quantization is substantially difficult, FP4DiT still provides the best results. Further, due to space constraints, we
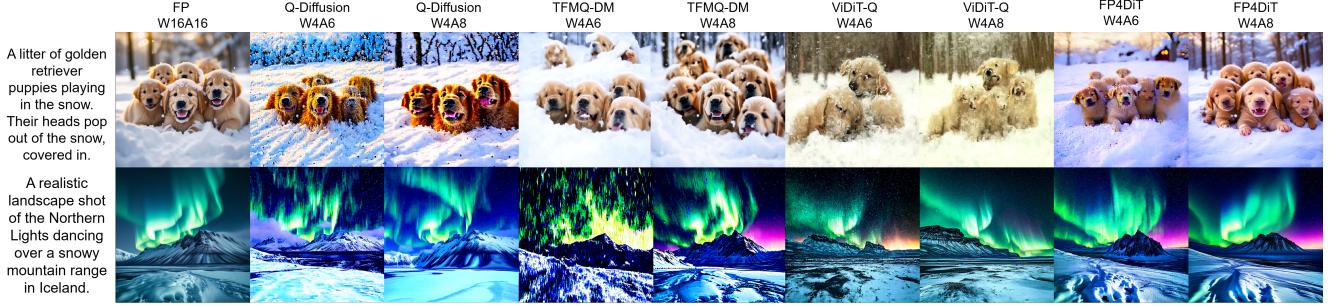
Figure 7. PixArt-α (up) and PixArt-Σ (down) images and comparison between FP4DiT and related work. Best viewed in color.

Figure 8. Hunyuan image comparison. Note the 'white background' detail for FP4DiT. Best viewed in color.

Model	Method	Precision	Preference(%) ↑
Hunyuan	Q-Diffusion	W4A8	6.58
	TFMQ-DM	W4A8	36.84
	FP4DiT (ours)	W4A8	**56.58**
	Q-Diffusion	W4A6	32.84
	TFMQ-DM	W4A6	28.36
	FP4DiT (ours)	W4A6	**38.81**

Table 3. User preference study between FP4DiT and baseline methods on Hunyuan DiT with W4A8 and W4A6 quantization.

report performance on each individual HPSv2 category as well as other metrics [13] in the in the supplementary. Overall though, Table 2 demonstrates the efficacy of FP4DiT in term of quantitative T2I compared to prominant baseline approaches.

Next, Figure 7 provides qualitative image samples on the PixArt models. Note the higher quality of the images generated by FP4DiT at both the A8 and A6 levels. Specifically, the puppies have more realistic detail and the generated image more closely aligns with the W16A16 model. This is especially true for the PixArt-Σ sample images, where the FP4DiT show detailed, but not blurry northern lights while maintaining detail on the snowy landscape in the foreground.

Further, Figure 8 provides image results on Hunyuan, where we again note the detail present in the FP4DiT image, while the baseline approaches are much noisier and have yellow backgrounds which are not prompt-adherent. Finally, additional visualization results can be found in the Supplementary Materials.

4.2.1. User Preference Study

We conduct a preference study to qualitatively compare FP4DiT to baseline approaches. Specifically, we consider 75 random prompts and 15 human participants. Each participant is presented with a prompt and the corresponding image generated by Hunyuan when quantized to W4A8 or W4A6 precision by FP4DiT, Q-Diffusion and TFMQ-DM. We record the percentage that each approach is selected as the preferred image and report the results in Table 3. At A8 precision FP4DiT clearly outperforms the other meth-

ods with a majority of the images being favored, while it also obtains over one third of the votes at A6 precision.

4.3. Ablation Studies

We conduct ablation studies on the PixArt-α model to verify the contribution of each component of FP4DiT. The experiment settings are consistent with Section 4.2 unless specified. Additional ablation studies can be found in the Supplementary Materials.

4.3.1. Effect of Weight Quantization

To evaluate the effectiveness of our weight quantization method, we perform an ablation study on weight-only quantization, e.g. W4A16. As depicted in Table 4, our method progressively improves the weight quantization: Initially, directly applying FPQ to Q-Diffusion results in a significant degradation. Our research then reveals that group quantization is necessary for FPQ. Notably, using the original AdaRound on top of FPQ impedes its effectiveness. Subsequently, our sensitive-aware mixed format FPQ (E3M0 in pointwise linear) further improves the post-quantization performance. Eventually, our scale-aware AdaRound advances the boundaries of optimal performance by considering the multi-scale nature of FP weight reconstruction.

4.3.2. Effect of Scale-Aware AdaRound

To further verify our scale-aware AdaRound for FP quantization, we compare our scale-aware AdaRound to the origin AdaRound with INT quantization in the W4A16 setting. Note that the calibration budget is crucial for the perfor-

Method	Precision	HPSv2 ↑
Full Precision	W16A16	31.01
Q-Diffusion	W4A16	25.22
Q-Diffusion-FPQ	W4A16	8.78
+ Group Quant	W4A16	25.05
+Scale-Aware AdaRound	W4A16	26.44
+Sensitive-aware FF Quant.	W4A16	28.21

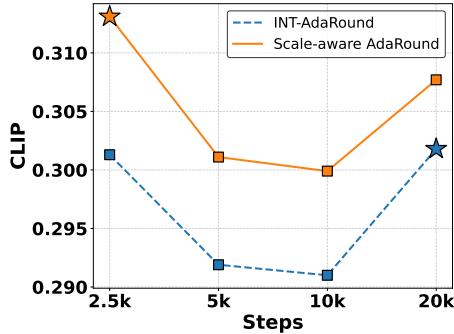Table 4. The effect of different methods proposed for weight quantization on W4A16 PixArt-α.

Figure 9. The performance of different AdaRound methods with different calibration budgets on W4A16 PixArt-α quantization generating 1k 512×512 images using COCO prompts. The stars indicate the optimal budget for INT and scale-aware AdaRound. The scale-aware AdaRound achieves an 8-fold reduction in calibration computation cost.

Model	Precision	Model Size (MB)↓	TBOPS↓
	W16A16	610.86	35.72
	W8A8	305.53	8.938
PixArt-α	W4A8	152.87	4.474
	W4A6	152.87	3.358
	W4A8-G128 (ours)	158.59	4.474
	W4A6-G128 (ours)	158.59	3.358

Table 5. The comparison of model size and latency of different Precision on PixArt-α. G128 denotes group-wise weight quantization with a group size of 128. Lower is better.

GPU	INT8	FP8	FP6	FP4
RTX 4090 [37]	660.6 TOPS	660.6 TFLOPS	–	–
H100 [39]	1979 TOPS	1979 TFLOPS	–	–
RTX 5090 [37]	838 TOPS	838 TFLOPS	838 TFLOPS	1676 TFLOPS
HGX B100 [38]	56 POPS	56 PFLOPS	56 PFLOPS	112 PFLOPS
HGX B200 [38]	72 POPS	72 PFLOPS	72 PFLOPS	144 PFLOPS

Table 6. GPU throughput rates for different low-bit datatype formats. Horizontal line demarcates older Ada Lovelace/Hopper GPUs from state-of-the-art Blackwell series. Older series don't support FP6 and FP4.

mance of weight reconstruction. BRECQ [24] uses 20k as default and this setting is inherited by prior DM quantization research like Q-Diffusion and TFMQ-DM. Thus, we configured calibration budgets at {2.5k, 5k, 10k, 20k} to ensure a fair comparison. Figure 9 outlines the CLIP of different AdaRound methods on PixArt-α under different calibration budgets. Scale-aware AdaRound achieving optimal budget with 8 times fewer calibration steps than INT AdaRound, highlights its effectiveness in reducing calibration costs without compromising reconstruction quality.

4.4. Hardware Cost Comparison

We compare the hardware cost of the quantized FP4DiT model with the full-precision model using two metrics: model size and Bit-Ops (BOPs) [11]. Specifically, model size is the disk space required to store the model checkpoint weights and scales, while BOPs is a quantization-aware extension of the MACs [4, 32] metric which measures the compute cost of a neural network forward pass.

Table 5 outlines our results. We see that the group-wise weight quantization only brings moderate overhead hence our method still substantially reduces the model size and BOPs computational complexity.

Finally, in terms of computational throughput, Table 6

shows the computation rate of different INT and FP formats across different Nvidia GPUs. Although the floating-point math is typically more complex, the state-of-the-art Nvidia "Blackwell" GPUs like the 5090, B100 and B200 are designed to handle INT and FP at the same computational throughput as shown, which ensures that FP-based quantization does not introduce additional computational overhead. Leveraging this, our FPQ method can achieve superior performance without sacrificing computational efficiency compared to INT quantization, making it an effective alternative. Although FP6 shares the same compute throughput as FP8, it brings memory saving (25% smaller tensor) which is essential for low-bit quantization, as transformers often become memory-bound [54] during token generation. As a result, FP6 enhances DiT's inference efficiency.

5. Conclusion

In this paper, we propose FP4DiT, a PTQ method that achieves W4A6 and W4A8 quantization on T2I DiT using FPQ. We use a mixed FP formats strategy based on the special structure of DiT and propose scale-aware AdaRound to enhance the weight quantization for FPQ. We analyze the difference between the activation of U-Net DM and DiT and apply token-wise online activation quantization based on the findings. Our experiments demonstrate the superior performance of FP4DiT compared to other quantization methods on the quantative HPSv2 benchmark, MS-COCO dataset and qualitative visualization comparison at minimal hardware cost.

References

[1] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 2

[2] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-Σ: Weak-to-strong training of diffusion transformer for 4k text-to-image generation, 2024. 1, 2, 6

[3] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Zhongdao Wang, James T. Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 1, 2, 6

[4] Xiaojie Chu, Liangyu Chen, and Wenqing Yu. Nafssr: Stereo image super-resolution using nafnet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1239–1248, 2022. 8

[5] 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. 1

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[7] 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. 1, 2

[8] Weilun Feng, Haotong Qin, Chuanguang Yang, Zhulin An, Libo Huang, Boyu Diao, Fei Wang, Renshuai Tao, Yongjun Xu, and Michele Magno. Mpq-dm: Mixed precision quantization for extremely low bit diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025. 1, 2

[9] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022. 6

[10] 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. 2, 5

[11] 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. 5, 8

[12] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 5, 6, 2

[13] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 7, 2

[14] Yushi Huang, Ruihao Gong, Jing Liu, Tianlong Chen, and Xianglong Liu. Tfmq-dm: Temporal feature maintenance quantization for diffusion models. *arXiv preprint arXiv:2311.16503*, 2023. 1, 2

[15] 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. 2, 5, 6

[16] 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. 2

[17] William Kahan. Ieee standard 754 for binary floating-point arithmetic. *Lecture Notes on the Status of IEEE*, 754(94720-1776):11, 1996. 3

[18] Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, et al. A study of bfloat16 for deep learning training. *arXiv preprint arXiv:1905.12322*, 2019. 3

[19] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018. 2

[20] Andrey Kuzmin, Mart Van Baalen, Yuwei Ren, Markus Nagel, Jorn Peters, and Tijmen Blankevoort. Fp8 quantization: The power of the exponent. *Advances in Neural Information Processing Systems*, 35:14651–14662, 2022. 1, 2

[21] Black Forest Labs. flux. 2

[22] Joonhyung Lee, Jeongin Bae, Byeongwook Kim, Se Jung Kwon, and Dongsoo Lee. To fp8 and back again: Quantifying the effects of reducing precision on llm training stability. *arXiv preprint arXiv:2405.18710*, 2024. 3

[23] 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. 1, 2, 4, 5, 6, 3

[24] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. Brecq: Pushing the limit of post-training quantization by block reconstruction. *arXiv preprint arXiv:2102.05426*, 2021. 1, 2, 4, 8

[25] Zhimin Li, Jianwei Zhang, Qin Lin, Jiangfeng Xiong, Yanxin Long, Xinchi Deng, Yingfang Zhang, Xingchao Liu, Minbin Huang, Zedong Xiao, Dayou Chen, Jiajun He, Jiahao

Li, Wenyue Li, Chen Zhang, Rongwei Quan, Jianxiang Lu, Jiabin Huang, Xiaoyan Yuan, Xiaoxiao Zheng, Yixuan Li, Jihong Zhang, Chao Zhang, Meng Chen, Jie Liu, Zheng Fang, Weiyan Wang, Jinbao Xue, Yangyu Tao, Jianchen Zhu, Kai Liu, Sihuan Lin, Yifu Sun, Yun Li, Dongdong Wang, Mingtao Chen, Zhichao Hu, Xiao Xiao, Yan Chen, Yuhong Liu, Wei Liu, Di Wang, Yong Yang, Jie Jiang, and Qinglin Lu. Hunyuan-dit: A powerful multi-resolution diffusion transformer with fine-grained chinese understanding, 2024. 2, 6

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. In *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. 5, 6

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2

[28] Shih-yang Liu, Zechun Liu, Xijie Huang, Pingcheng Dong, and Kwang-Ting Cheng. Llm-fp4: 4-bit floating-point quantized transformers. *arXiv preprint arXiv:2310.16836*, 2023. 2

[29] Wenxuan Liu and Saiqian Zhang. Hq-dit: Efficient diffusion transformer with fp4 hybrid quantization. *arXiv preprint arXiv:2405.19751*, 2024. 2

[30] Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*, 2017. 4

[31] Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui 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

[32] Keith G. Mills, Di Niu, Mohammad Salameh, Weichen Qiu, Fred X. Han, Puyuan Liu, Jialin Zhang, Wei Lu, and Shangling Jui. Aio-p: Expanding neural performance predictors beyond image classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8):9180–9189, 2023. 8

[33] Keith G. Mills, Mohammad Salameh, Ruichen Chen, Wei Hassanpour, Negar Lu, and Di Niu. Qua^{2}sedimo: Quantifiable quantization sensitivity of diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025. 1, 2

[34] 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. 1, 2, 4

[35] 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. 2

[36] Yury Nahshan, Brian Chmiel, Chaim Baskin, Evgenii Zheltonozhskii, Ron Banner, Alex M Bronstein, and Avi

Mendelson. Loss aware post-training quantization. *Machine Learning*, 110(11):3245–3262, 2021. 1

[37] NVIDIA. Nvidia rtx blackwell gpu architecture, 2024. Accessed: 2025-3-07. 8

[38] NVIDIA. Nvidia blackwell architecture technical overview, 2024. Accessed: 2024-11-07. 8

[39] NVIDIA. Nvidia h100 tensor core gpu architecture overview, 2024. Accessed: 2024-11-07. 8

[40] Gunho Park, Baeseong Park, Minsub Kim, Sungjae Lee, Jeonghoon Kim, Beomseok Kwon, Se Jung Kwon, Byeongwook Kim, Youngjoo Lee, and Dongsoo Lee. Lut-gemm: Quantized matrix multiplication based on luts for efficient inference in large-scale generative language models. *arXiv preprint arXiv:2206.09557*, 2022. 6, 2

[41] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 1, 2

[42] 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. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. 1

[43] Enrico Reggiani, Renzo Andri, and Lukas Cavigelli. Flexsfu: Accelerating dnn activation functions by non-uniform piecewise approximation. In *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2023. 3

[44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022. 1, 2

[45] 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. 1, 2, 5

[46] Haihao Shen, Naveen Mellempudi, Xin He, Qun Gao, Chang Wang, and Mengni Wang. Efficient post-training quantization with fp8 formats. *Proceedings of Machine Learning and Systems*, 6:483–498, 2024. 1, 3

[47] Junhyuk So, Jungwon Lee, Daehyun Ahn, Hyungjun Kim, and Eunhyeok Park. Temporal dynamic quantization for diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2

[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. *Advances in Neural Information Processing Systems*, 37, 2025. 2

[49] Mart van Baalen, Andrey Kuzmin, Suparna S Nair, Yuwei Ren, Eric Mahurin, Chirag Patel, Sundar Subramanian, Sanghyuk Lee, Markus Nagel, Joseph Soriaga, et al. Fp8 versus int8 for efficient deep learning inference. *arXiv preprint arXiv:2303.17951*, 2023. 4

[50] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj,

Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 6, 2

[51] Guibin Wang, YiSong Lin, and Wei Yi. Kernel fusion: An effective method for better power efficiency on multithreaded gpu. In *2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, pages 344–350. IEEE, 2010. 5

[52] Xiaoshi Wu, Yiming Hao, Keqiang Sun, Yixiong Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score v2: A solid benchmark for evaluating human preferences of text-to-image synthesis. *arXiv preprint arXiv:2306.09341*, 2023. 2, 6

[53] Xiaoxia Wu, Zhewei Yao, and Yuxiong He. Zeroquant-fp: A leap forward in llms post-training w4a8 quantization using floating-point formats. *arXiv preprint arXiv:2307.09782*, 2023. 2, 5, 6

[54] Haojun Xia, Zhen Zheng, Xiaoxia Wu, Shiyang Chen, Zhewei Yao, Stephen Youn, Arash Bakhtiari, Michael Wyatt, Donglin Zhuang, Zhongzhu Zhou, et al. {Quant-LLM}: Accelerating the serving of large language models via {FP6-Centric}{Algorithm-System}{Co-Design} on modern {GPUs}. In *2024 USENIX Annual Technical Conference (USENIX ATC 24)*, pages 699–713, 2024. 8

[55] 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. 1, 2

[56] Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. Sana: Efficient high-resolution image synthesis with linear diffusion transformer, 2024. 2

[57] 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. 1, 2, 5, 6

[58] Tianchen Zhao, Tongcheng Fang, Enshu Liu, Wan Rui, Widyadewi Soedarmadji, Shiyao Li, Zinan Lin, Guohao Dai, Shengen Yan, Huazhong Yang, et al. Vidit-q: Efficient and accurate quantization of diffusion transformers for image and video generation. *arXiv preprint arXiv:2406.02540*, 2024. 1, 2, 6

vb

FP4DiT: Towards Effective Floating Point Quantization for Diffusion Transformers

Supplementary Material

A. Proof for Scale-Aware AdaRound

AdaRound [34] and BRECQ [24] use gradient descent to update the rounding mask. According to Eqs. 4, 5 and 6, ignoring the regularizer $f_{\text{reg}}(\mathbf{V})$, we have following theorem which provides theoretical evidence that the origin AdaRound is imbalance across different scales s.

Theorem 1. *Let s be the quantization scale corresponding to the rounding mask V. Then for gradient descent, given as $\mathbf{V}_{n+1} = \mathbf{V}_n - \alpha \nabla F(\mathbf{V}_n)$, the subtraction $\nabla F(\mathbf{V}_n)$ is dependent on the scalar s.*

Proof. Refers to Eqs. 4, 5 and gives:

$$\nabla F(\mathbf{V}_n) = \nabla_{\widetilde{W}} \|W\mathbf{x} - \widetilde{W}\mathbf{x}\|_F^2 \cdot s \cdot \frac{\partial h(\mathbf{V}_n)}{\partial \mathbf{V}_n} \quad (9)$$

Refers to Eq. 6 and gives:

$$\frac{\partial h(\mathbf{V})}{\partial \mathbf{V}} = (\zeta - \gamma) \cdot \sigma(\mathbf{V}) \cdot (1 - \sigma(\mathbf{V})) \quad (10)$$

Therefore, combining Eq. 9 and 10, the subtraction $\nabla F(\mathbf{V}_n)$ is scaled by s. □

This theorem indicates that the imbalance gradient descent occurs if applying origin AdaRound to FPQ, as shown in Figure 4b.

In the section 3, we propose a scale-aware version of AdaRound in Eq. 7 and 8. Ignoring the regularizer f_{reg}, we have the following theorem which provides theoretical evidence that the scale-aware AdaRound has balanced gradient descent's update across different scales s.

Theorem 2. *Let s be the quantization scale corresponding to the rounding mask V'. Then for gradient descent, given as $\mathbf{V'}_{n+1} = \mathbf{V'}_n - \alpha \nabla F(\mathbf{V'}_n)$, the subtraction $\nabla F(\mathbf{V'}_n)$ is independent of the scalar s.*

Proof. The learning objective doesn't change for scale-aware AdaRound. Hence, from Eq. 4 and 7, we give:

$$\nabla F(\mathbf{V'}_n) = \nabla_{\widetilde{W}} \|W\mathbf{x} - \widetilde{W}\mathbf{x}\|_F^2 \cdot s \cdot \frac{\partial h'(\mathbf{V'}_n)}{\partial \mathbf{V'}_n} \quad (11)$$

Refer to Eq. 8 and give:

$$\frac{\partial h'(\mathbf{V'})}{\partial \mathbf{V'}} = \frac{(\zeta - \gamma)}{s} \cdot \sigma(\mathbf{V'}) \cdot (1 - \sigma(\mathbf{V'})) \quad (12)$$

Combining Eq. 11 and 12, we get:

$$\nabla F(\mathbf{V'}_n) = \nabla_{\widetilde{W}} \|W\mathbf{x} - \widetilde{W}\mathbf{x}\|_F^2 \cdot (\zeta - \gamma) \cdot \sigma(\mathbf{V'}) \cdot (1 - \sigma(\mathbf{V'}))$$

This result shows that subtraction $\nabla F(\mathbf{V'}_n)$ is independent of scale s. □
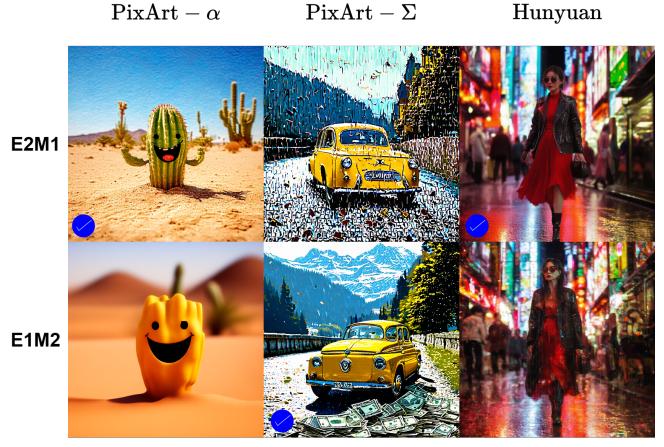

Figure 10. E2M1 (up) and E1M2 (down) min-max FPQ visualization results on PixArt and Hunyuan DiT. E2M1 is preferred by PixArt-α and Hunyuan, while E1M2 is preferred by PixArt-Σ.

Therefore, the gradient descent is balanced and normalized among different scale s with our scale-aware AdaRound, as depicted in Figure 4c.

A.1. Additional Ablation Studies

In addition to the ablation studies in the main text, we include multiple ablation studies here to further demonstrate the design of FP4DiT.

A.1.1. Effect of FP Format.

Recall the mixed format strategy for FPQ in FP4DiT: we apply E3M0, which allocates more value closer to the GELU sensitive interval, to the first pointwise linear layer and use a unified FP format from E2M1 and E1M2 for the rest layers. To choose a better one between E2M1 and E1M2 while avoiding data leakage, instead of quantizing FP4DiT with two formats and selecting the better one based on the FID and CLIP, we only employ the basic min-max quantization scheme (without AdaRound technique thereby avoiding the use of calibration data) and leave the activation un-quantize (e.g. W4A16). We use the PixArt prompts to generate images and apply user preference studies to determine the format.

Figure 10 shows the visualization results of E2M1 and E1M2 FPQ. For the PixArt-α, E2M1 demonstrates better suitability, as the cactus in E1M2 loses its texture, resulting in a mismatch between the image and the prompt. For PixArt-Σ and Hunyuan, inappropriate FP format causes

Model	Group Size	FID ↓
PixArt-α	128	96.58
	256	100.60
	512	113.78
	1024	174.38

Table 7. The quantization results for different group sizes with PixArt-α.

Model	Cali. Size	Cali. Step	Weight Format	Act. Format
PixArt-α	128	2500	E2M1	E2M3/E3M4
PixArt-Σ	128	2500	E1M2	E2M3/E3M4
Hunyuan	64	2500	E2M1	E2M3/E3M4

Table 8. Quantization calibration hyperparameters for FP4DiT.

Algorithm 1 MinMax quantization for FP format

Require: Full-precision array A_{FP}, number of bits n, number of exponent bits n_e, number of mantissa bits n_m, clipping value $maxval$

$A_{\text{abs}} \leftarrow \text{abs}(A_{\text{FP}})$
$bias \leftarrow 2^{n_e} - \log_2(A_{\text{abs}}) + \log_2(2 - 2^{-n_m}) - 1$
$A_{\text{clip}} \leftarrow \min(\max(A_{\text{FP}}, -maxval), maxval)$
$S_{\text{log}} \leftarrow \text{clamp}(\lfloor \log_2(\text{abs}(A_{\text{clip}})) \rfloor + bias)), 1)$
$S \leftarrow 2.0^{(S_{\text{log}} - n_m - bias)}$
$result \leftarrow \text{round-to-nearest}(A_{\text{clip}}/S) \times S$
return $result$

---

noise on the generated image, leading to suboptimal performance. In conclusion, it is straightforward and unambiguous to determine the unified FP format based on these visualization results.

A.1.2. Effect of Group Size.

We compare the group-wise weight quantization result with different group sizes in Table 7. Decreasing the group size g for weight quantization consistently improves the quantization performance down to $g = 128$. According to [40], group size g such as 128 can result in substantial improvement while maintaining low latency.

B. Extended Experimental Settings

B.1. Floating Point Quantization Scheme

Since Floating Point Quantization (FPQ) is not as straightforward as INT quantization, there hasn't been a simple and unified algorithm for performing FPQ yet. In this paper, we apply Algorithm 1 from [20] to perform the FPQ. Unlike [20] learning the clipping value $maxval$ and bit allocations between mantissa and exponent part, we use the absolute maximum of the tensor as $maxval$ and perform FPQ with a predetermined FP format.

In addition to the FPQ algorithm and the group-wise/token-wise quantization, we provide our quantization hyperparameters in Table 8. Note that the calibration size refers to the number of images we sampled from MS-COCO. For each image, we sample its input latent noise across 20 timesteps (50 for Hunyuan) as the calibration data for AdaRound. The activation FP format is for W4A6/W4A8 respectively.

B.2. Baseline implementation

In this paper, we compare FP4DiT with three baseline: Q-Diffusion [23], TFMQ-DM [15] and ViDiT-Q [58]. Their implementation details are enumerated as follows:

1. Q-Diffusion: Q-Diffusion is developed on DDIM, LDM, and the original Stable Diffusion v1.4 repository but does not provide native support for any DiT-based DM. Thus, we integrate the Hugging Face Diffusers [50] API with Q-Diffusion to provide support for PixArt-α and PixArt-Σ. Q-Diffusion has two techniques tailored for Diffusion Models (DM): time step-aware calibration and shortcut-splitting quantization. However, the shortcut-splitting is unworkable for the PixArt model as there are no U-Net shortcuts in either PixArt DiT. As for the activation quantization, we replaced their temporal-aware calibration quantization with online token-wise quantization for better performance and fair comparison.

2. TFMQ-DM: TFMQ-DM is also originally developed for U-Net DMs so we integrate the Hugging Face Diffusers API. However, its Temporal Information Block (TIB) consolidates all *embedding layers* and *time embed* into a unified quantization block, which is based on the unique structure of U-Net. Our solution is to replace the TIB in TFMQ-DM with the time embedding module in the DiT models. Similar to Q-Diffusion, online activation is also applied to the TFMQ-DM.

3. ViDiT-Q: ViDiT-Q supports the quantization PixArt-α and PixArt-Σ. ViDiT-Q uses a mix-precision quantization strategy: For certain layers, ViDiT-Q quantizes them with higher precision like 8 bits, or skips their quantization (16 bits). In our experiments, we use it as W4 and perform the comparison. Besides, they employ online activation quantization, which can be fairly compared to FP4DiT.

B.3. Evaluation Settings

For CLIP score [12], we apply the openai-clip [1] library to measure the CLIP score between prompts and generated images. We employ ViT-B/32 as the CLIP model. To measure HPSv2 [52] score, we generate 3.2k images across the four categories (800 images per category) and use the provided human preference predictor to estimate the performance. For the Fréchet Inception Distance (FID) [13]

---

[1] https://github.com/openai/CLIP

Benchmark			HPSv2					MS-COCO	
Model	Method	Precision	Animation↑	Concept-art↑	Painting↑	Photo↑	Average↑	FID↓	CLIP↑
PixArt-α	Full Precision	W16A16	32.56	31.06	30.76	29.67	31.01	50.42	0.3075
	Q-Diffusion	W4A8	24.18	23.43	22.91	22.15	23.17	70.34	0.3017
	TFMQ-DM	W4A8	*27.88*	*26.03*	*25.14*	*24.91*	*25.99*	72.41	*0.3066*
	ViDiT-Q	W4A8	17.93	17.35	16.81	17.59	17.42	*49.04*	0.2900
	FP4DiT (ours)	W4A8	**28.63**	**26.79**	**27.59**	**26.72**	**27.43**	**46.18**	**0.3076**
	Q-Diffusion	W4A6	12.63	13.39	13.32	10.65	12.50	92.24	0.2868
	TFMQ-DM	W4A6	*24.02*	**23.36**	22.72	*23.04*	23.29	75.24	*0.3015*
	ViDiT-Q	W4A6	16.71	16.28	16.23	16.56	16.44	*57.81*	0.2827
	FP4DiT (ours)	W4A6	**24.57**	*23.08*	**23.30**	**23.26**	**23.55**	**57.17**	**0.3031**
PixArt-Σ	Full Precision	W16A16	33.07	31.58	31.54	30.49	31.67	47.97	0.3139
	Q-Diffusion	W4A8	*27.30*	26.11	26.06	**25.06**	*26.13*	50.63	*0.3050*
	TFMQ-DM	W4A8	24.95	23.59	23.19	22.73	23.62	77.82	0.2975
	ViDiT-Q	W4A8	27.10	**26.45**	*26.24*	24.49	26.07	*50.43*	0.2562
	FP4DiT (ours)	W4A8	**27.95**	*26.29*	**26.16**	*24.67*	**26.27**	**48.73**	**0.3064**
	Q-Diffusion	W4A6	*24.07*	22.42	22.47	21.72	22.67	*60.25*	*0.3027*
	TFMQ-DM	W4A6	19.30	18.46	18.85	17.50	18.53	169.24	0.2346
	ViDiT-Q	W4A6	24.84	*23.23*	*23.48*	*21.94*	*23.37*	106.22	0.2425
	FP4DiT (ours)	W4A6	**26.91**	**25.55**	**25.22**	**23.93**	**25.40**	**58.59**	**0.3040**
Hunyuan	Full Precision	W16A16	33.72	31.84	31.52	31.24	32.08	68.41	0.3102
	Q-Diffusion	W4A8	26.00	24.74	24.84	24.26	24.96	90.16	0.3006
	TFMQ-DM	W4A8	*28.77*	*27.63*	**27.67**	*26.15*	*27.56*	*84.34*	*0.3075*
	FP4DiT (ours)	W4A8	**28.96**	**27.75**	*27.47*	**26.94**	**27.78**	**79.51**	**0.3102**
	Q-Diffusion	W4A6	*14.90*	*14.83*	*15.32*	14.24	*14.83*	264.75	0.2277
	TFMQ-DM	W4A6	13.73	13.31	13.16	**14.51**	13.68	*256.42*	*0.2520*
	FP4DiT (ours)	W4A6	**15.23**	**14.94**	**15.57**	*14.27*	**15.00**	**248.11**	**0.2562**

Table 9. Quantitative evaluation results for PixArt-α, PixArt-Σ and Hunyuan in terms of FID and CLIP score. Specifically, for each configuration, we generate 10k images (3k for Hunyuan) using COCO 2017 validation set prompts. Best and second best results in **bold** and *italics*, respectively.

measurement, we apply clean-fid [2] library to measure the FID between generated images and ground-truth images.

B.4. Hardware and Software Resources

We execute our FP4DiT and baseline experiments on two rack servers. The first is equipped with 2 Nvidia A100 80 GPUs, an AMD EPYC-Rome Processor and 512GB RAM. The second server has 8 Nvidia V100 32GB GPUs, an Intel Xeon Gold GPU and 756GB RAM.

Our code is running under Python 3 using Anaconda virtual environments and open-source repository forks based on Q-Diffusion [23]. We modified the code to implement our FP4DiT and enable the interface between Quantization scripts and the Hugging Face Diffusers [3] library. The hardware cost measurement is conducted using the ptflops [4] library. We provide a code implementation with README listing all necessary details and steps.

[2] https://github.com/GaParmar/clean-fid
[3] https://huggingface.co/docs/diffusers/en/index
[4] https://github.com/sovrasov/flops-counter.pytorch

C. Additional Quantization Results

In this section, we present the detailed HPSv2 scores and the FID and CLIP on the MS-COCO dataset. Table 9 enumerates all quantitative results for FP4DiT and other baselines on PixArt-α, PixArt-Σ and Hunyuan. Aligning with FP4DiT's superior HPSv2 score shown in Table 2, FP4DiT outperforms all baseline methods across different precisions and models in terms of FID and CLIP. Although FP4DiT doesn't lead under some HPSv2 categories, FP4DiT's score is close to the best thus achieve best average HPSv2 score across all tasks. The consistent superior results on HPSv2, FID and CLIP verify the effectiveness of FP4DiT.

D. Additional Visualization Results

In this section, we present the random samples derived from full-precision and W4A6/W4A8 PixArt-α, PixArt-Σ, and Hunyuan that are quantized by different baseline and FP4DiT. As depicted by Figures 11, 12, 13, 14, 15, and 16, FP4DiT generates results of impressive visual content. FP4DiT consistently shows superior performance across various DiT models. We list a detailed comparison between FP4DiT and other baselines as follows:

1. PixArt-α: On W4A8 (Fig. 11), FP4DiT generates more fine-grained images than all other baselines, such as the texture of macarons and waves. Besides, other baselines fail to generate a hedgehog while FP4DiT correctly depicts it. On W4A6 (Fig. 12), our method shows near W4A8 performance, while other baselines become noisy and lose details.

2. PixArt-Σ: On W4A8 (Fig. 13), there is almost no noise on FP4DiT's generated images, which demonstrates the improvement of our method. As for the image details, FP4DiT has a more natural yoga posture, a more fine-grained model face, and more detailed cat hair and whiskers. Similar to PixArt-α, FP4DiT's W4A6 (Fig. 14) images have the least noise and best image quality.

3. Hunyuan: On W4A8 (Fig. 15), FP4DiT's generated images more closely align with the full precision model. For example, in the first image, FP4DiT has the same color right face while Q-Diffusion and TMFQ-DM alter the face's color. In the third image, FP4DiT generates the most similar face decoration. In addition, FP4DiT also has the least noise on Hunyuan DiT. On W4A6 (Fig. 16), all methods generate blur images. However, the contour of images identified from FP4DiT's visualization results matches the full precision images, while other baselines fail to produce a contour or the contour is not prompt-adherent.
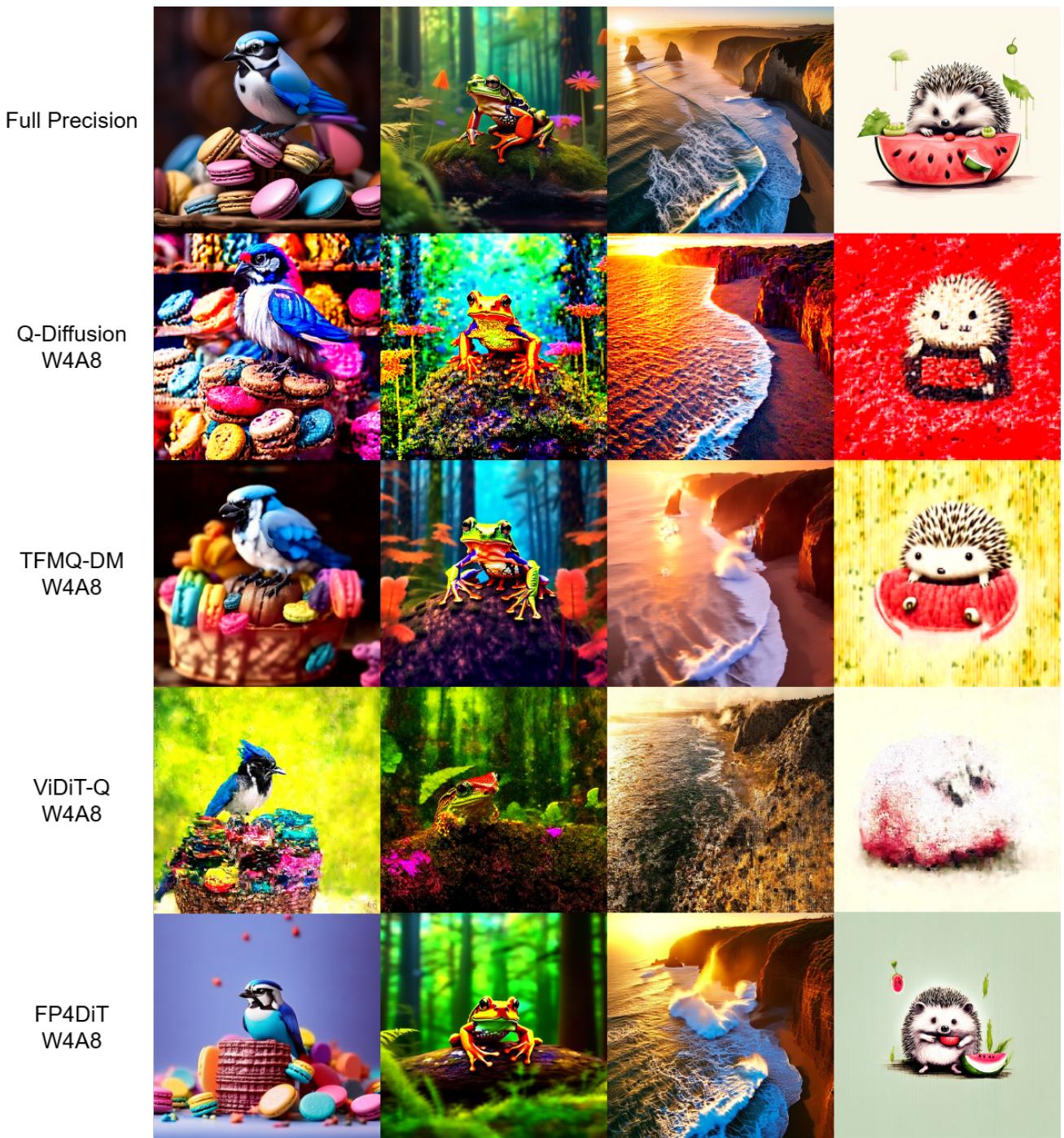
Figure 11. More visualization result for W4A8 PixArt-α. Prompts: 'A blue jay standing on a large basket of rainbow macarons.'; 'Frog, in forest, colorful, no watermark, no signature, in forest, 8k.'; 'Drone view of waves crashing against the rugged cliffs along Big Sur's Garay Point beach. The crashing blue waters create white-tipped waves, while the golden light of the setting sun illuminates the rocky shore.'; 'An ink sketch style illustration of a small hedgehog holding a piece of watermelon with its tiny paws, taking little bites with its eyes closed in delight. Photo of a lychee-inspired spherical chair, with a bumpy white exterior and plush interior, set against a tropical wallpaper.'
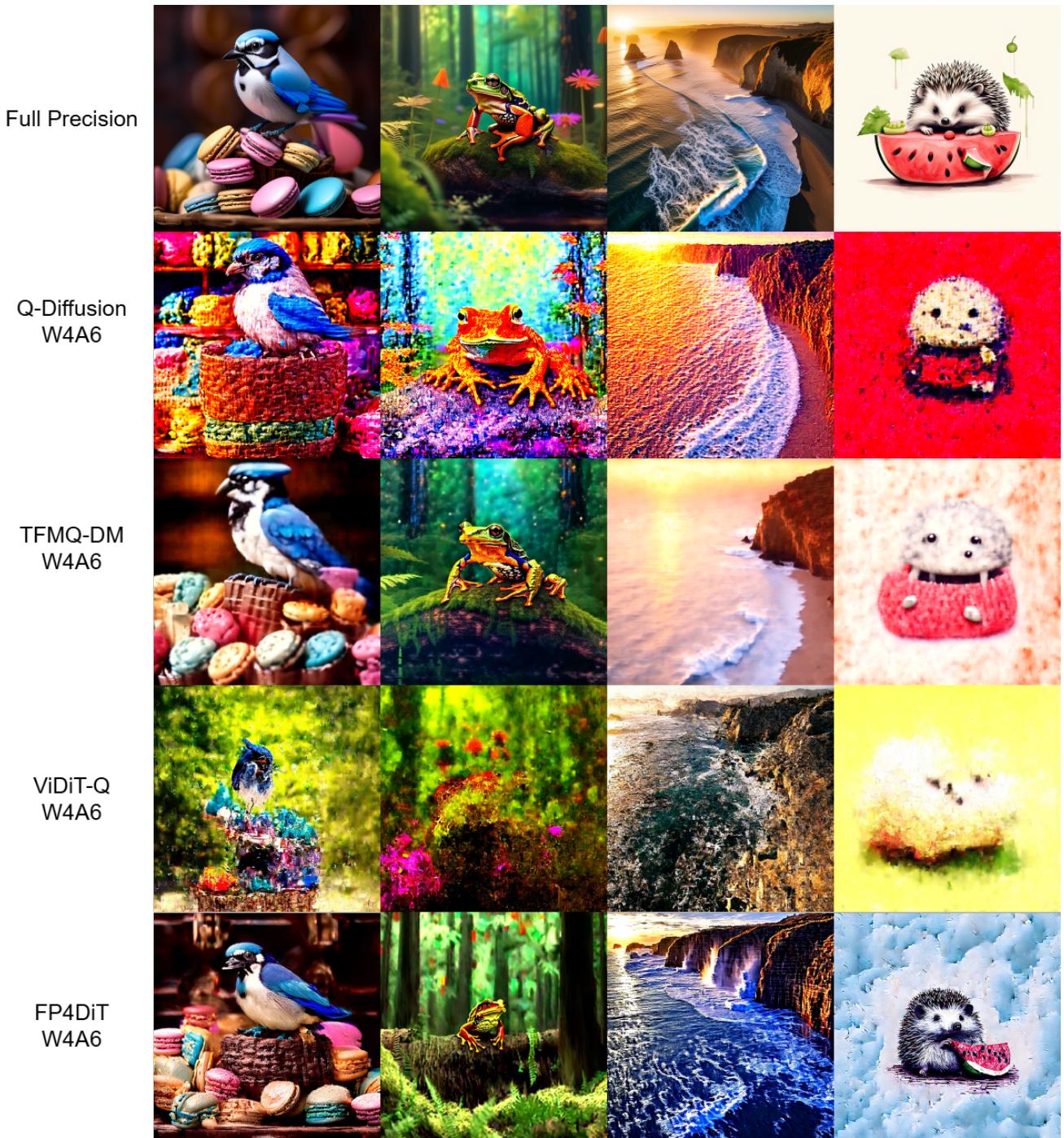
Figure 12. More visualization result for W4A6 PixArt-α. Prompts: 'A blue jay standing on a large basket of rainbow macarons.'; 'Frog, in forest, colorful, no watermark, no signature, in forest, 8k.'; 'Drone view of waves crashing against the rugged cliffs along Big Sur's Garay Point beach. The crashing blue waters create white-tipped waves, while the golden light of the setting sun illuminates the rocky shore.'; 'An ink sketch style illustration of a small hedgehog holding a piece of watermelon with its tiny paws, taking little bites with its eyes closed in delight. Photo of a lychee-inspired spherical chair, with a bumpy white exterior and plush interior, set against a tropical wallpaper.'

Figure 13. More visualization result for W4A8 PixArt-Σ. Prompts: 'A very attractive and natural woman, sitting on a yoka mat, breathing, eye closed, no make up, intense satisfaction, she looks like she is intensely relaxed, yoga class, sunrise, 35mm.'; 'Realistic oil painting of a stunning model merged in multicolor splash made of finely torn paper, eye contact, walking with class in a street.'; 'A cute orange kitten sliding down an aqua slide. happy excited. 16mm lens in front. we see his excitement and scared in the eye. vibrant colors. water splashing on the lens.'
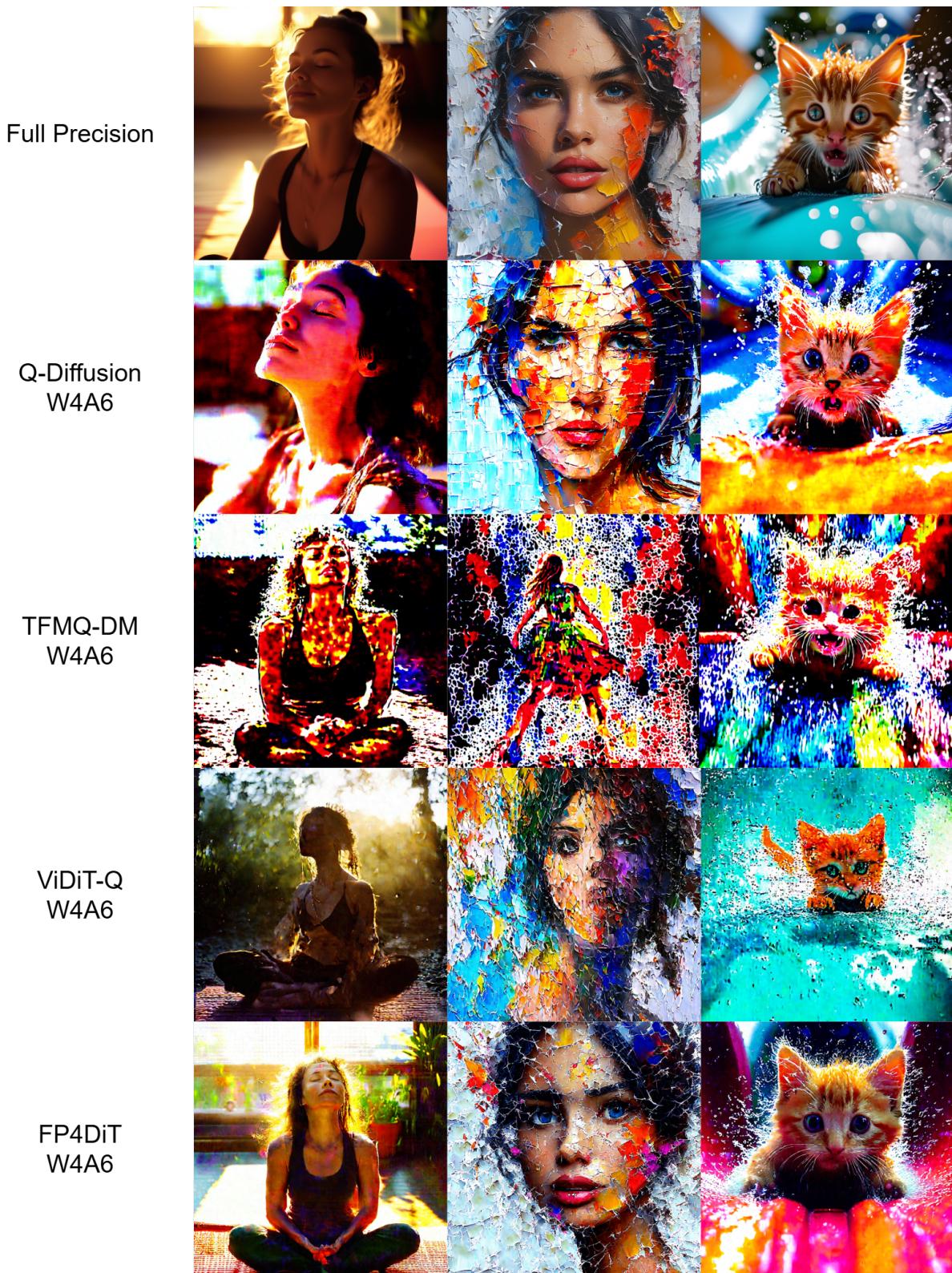
Figure 14. More visualization result for W4A6 PixArt-Σ. Prompts: 'A very attractive and natural woman, sitting on a yoka mat, breathing, eye closed, no make up, intense satisfaction, she looks like she is intensely relaxed, yoga class, sunrise, 35mm.'; 'Realistic oil painting of a stunning model merged in multicolor splash made of finely torn paper, eye contact, walking with class in a street.'; 'A cute orange kitten sliding down an aqua slide. happy excited. 16mm lens in front. we see his excitement and scared in the eye. vibrant colors. water splashing on the lens.'
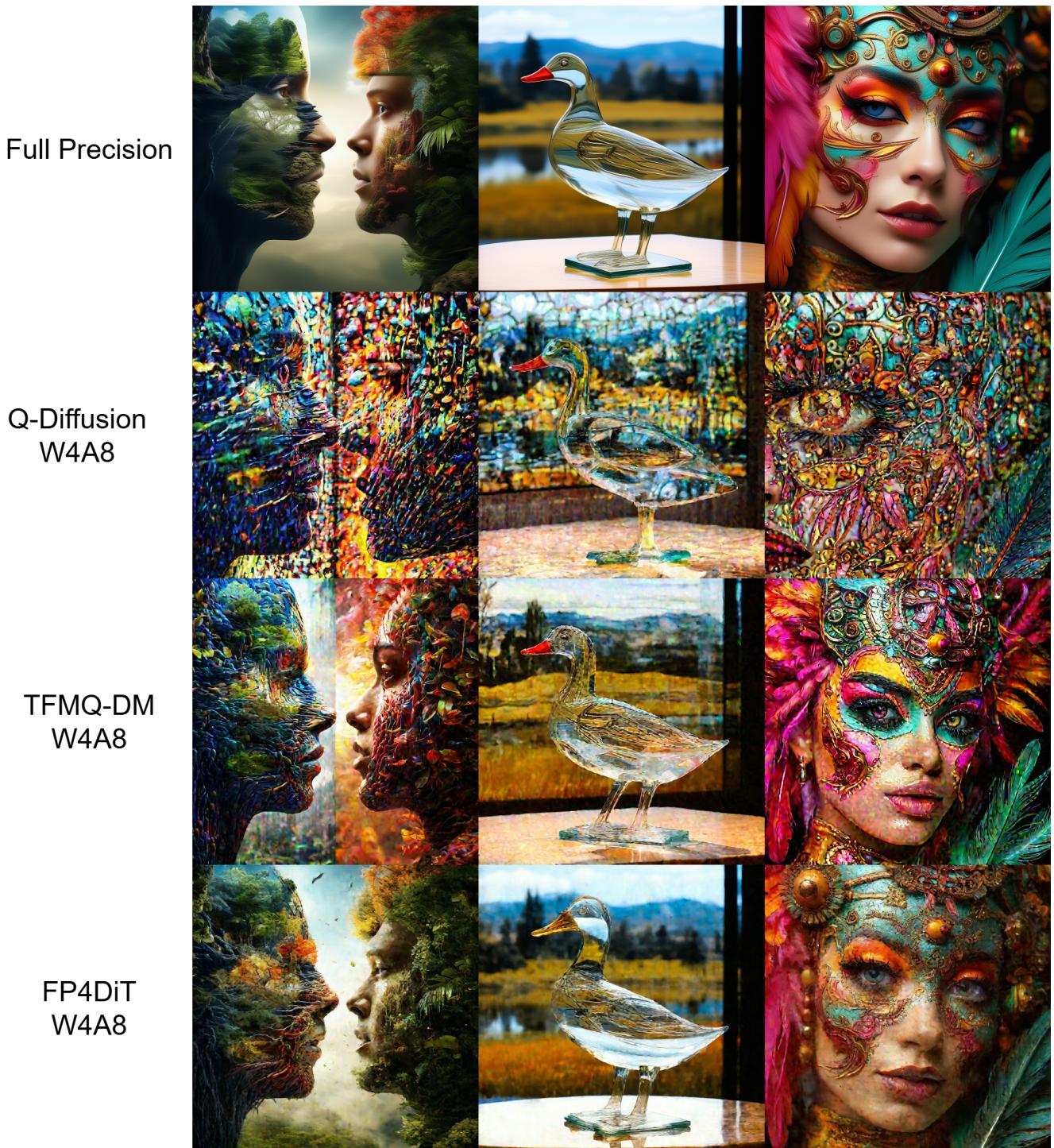
Figure 15. More visualization result for W4A8 Hunyuan. Prompts: 'nature vs human nature, surreal, UHD, 8k, hyper details, rich colors, photograph.'; 'A transparent sculpture of a duck made out of glass. The sculpture is in front of a painting of a landscape.'; 'Steampunk makeup, in the style of vray tracing, colorful impasto, uhd image, indonesian art, fine feather details with bright red and yellow and green and pink and orange colours, intricate patterns and details, dark cyan and amber makeup. Rich colourful plumes. Victorian style.'

9

Figure 16. More visualization result for W4A6 Hunyuan. Prompts: 'nature vs human nature, surreal, UHD, 8k, hyper details, rich colors, photograph.'; 'A transparent sculpture of a duck made out of glass. The sculpture is in front of a painting of a landscape.'; 'Steampunk makeup, in the style of vray tracing, colorful impasto, uhd image, indonesian art, fine feather details with bright red and yellow and green and pink and orange colours, intricate patterns and details, dark cyan and amber makeup. Rich colourful plumes. Victorian style.'