

# ViDiT-Q: Efficient and Accurate Quantization of Diffusion Transformers for Image and Video Generation

Tianchen Zhao<sup>1,2</sup>, Tongcheng Fang<sup>1,2</sup>, Enshu Liu<sup>1</sup>, Wan Rui<sup>1</sup>, Widyadewi Soedarmadji<sup>1</sup>, Shiyao Li<sup>1,2</sup>, Zinan Lin<sup>3</sup>, Guohao Dai<sup>2,4</sup>, Shengen Yan<sup>2</sup>, Huazhong Yang<sup>1</sup>, Xuefei Ning<sup>1†‡</sup>, Yu Wang<sup>1†</sup>

<sup>1</sup>Tsinghua University, <sup>2</sup>Infinigence AI, <sup>3</sup>Microsoft, <sup>4</sup>Shanghai Jiao Tong University

†Corresponding authors, ‡Project Advisor

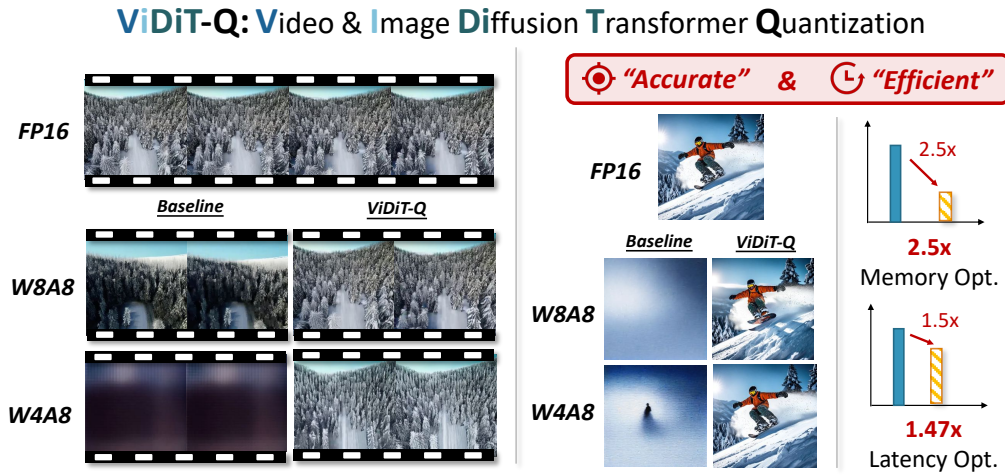


Figure 1: We introduce ViDiT-Q, a quantization method specialized for diffusion transformers used in image and video generation. ViDiT-Q achieves lossless W8A8 quantization and minimal visual quality degradation at W4A8, gaining 2.5x model size reduction and a 1.5x latency speedup.<sup>1</sup>

## Abstract

Diffusion transformers have exhibited remarkable performance in visual generation tasks, such as generating realistic images or videos based on textual instructions. However, larger model sizes and multi-frame processing for video generation lead to increased computational and memory costs, posing challenges for practical deployment on edge devices. Post-Training Quantization (PTQ) is an effective method for reducing memory costs and computational complexity. When quantizing diffusion transformers, we find that applying existing diffusion quantization methods designed for U-Net faces challenges in preserving quality. After analyzing the major challenges for quantizing diffusion transformers, we design an improved quantization scheme: ViDiT-Q (Video & Image Diffusion Transformer Quantization) to address these issues. Furthermore, we identify highly sensitive layers and timesteps hinder quantization for lower bit-widths. To tackle this, we improve ViDiT-Q with a novel metric-decoupled mixed-precision quantization method (ViDiT-Q-MP). We validate the effectiveness of ViDiT-Q across a variety of text-to-image and video models. While baseline quantization methods fail at W8A8 and produce unreadable content at W4A8, ViDiT-Q achieves lossless W8A8 quantization. ViDiT-Q-MP achieves W4A8 with negligible visual quality degradation, resulting in a 2.5x memory optimization and a 1.5x latency speedup.

<sup>1</sup>All videos in the Figures are provided in : <https://a-suozhang.xyz/viditq.github.io/>.

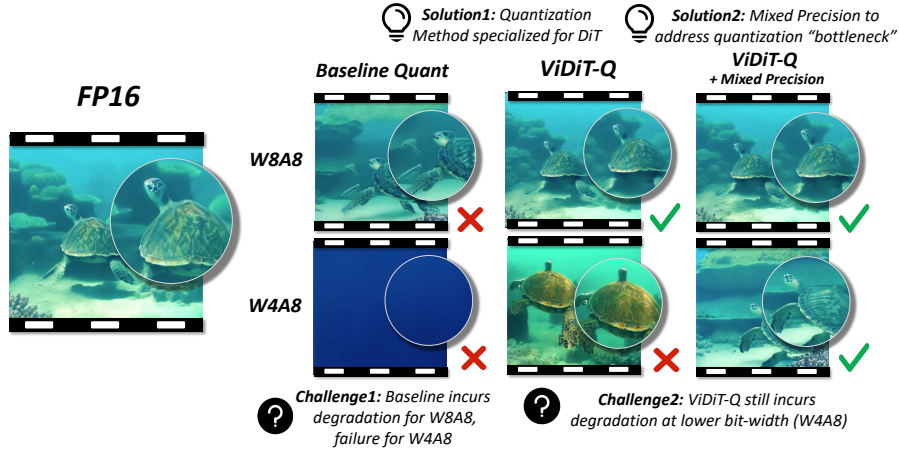


Figure 2: **The Challenges for existing diffusion quantization methods, and ViDiT-Q’s solutions.** ViDiT-Q introduced improved quantization scheme tailored for DiT to achieve lossless W8A8, and metric decoupled mixed precision tailored for video generation to mitigate degradation for W4A8.

## 1 Introduction

Diffusion Transformers [1, 44] and video generation tasks [22, 37, 53] have garnered significant research interest since the impressive performance of SORA [41]. However, the increasing model size poses challenges for application and deployment on edge devices. In the realm of video generation, processing multiple frames imposes a significant burden on both memory and computation. For example, the OpenSORA [22] model consumes over 10 GB of GPU memory to generate a single  $512 \times 512$  resolution video with only 16 frames, taking about 50 seconds on an Nvidia A100 GPU.

Model quantization [25, 38] has proven to be an effective compression method. By compressing high bit-width floating-point (FP) data into lower bit-width integers, the computational and memory costs of the model can be effectively reduced. While many prior studies [16, 27] have explored quantization for CNN (U-Net)-based diffusion models [49], we empirically observe challenges when quantizing Diffusion Transformers (DiTs) for image and video generation. Specifically, as illustrated in Fig. 2, the W8A8 quantized open-sora (STDiT [22]) model experiences notable quality degradation (the turtle has multiple fins), and the W4A16 quantized model produces single-colored blank frames.

By observing the data distribution (See Fig. 4), we conclude the primary issue with DiT quantization is adopting the same set of fixed quantization parameters for highly variant values in 4 levels (See Sec. 3.1): (1) Input-channel level, (2) Token level, (3) Timestep level, and (4) Classifier-free-guidance level. Correspondingly, we design an improved quantization scheme: “**ViDiT-Q**”, tailored for DiTs, to address this challenge. This scheme includes timestep-aware channel balancing, token-wise and dynamic quantization. ViDiT-Q enables W8A8 quantization without visual degradation.

Furthermore, in Fig. 2, when we continue to explore lower-bit quantization with uniform bit-widths (W6A6 and W4A8), notable degradation is witnessed. Through investigation, we discover that the quantization sensitivity varies significantly across layers and timesteps. Certain layers exhibit high sensitivity, acting as “bottlenecks” for quantization. To address this issue, inspired by prior mixed precision research [10, 12, 42, 62], we adopt higher bit-widths to “protect” these sensitive layers. When assigning proper bit-width for each layer, we find that simple Mean Square Error with FP-generated videos [68] cannot adequately distinguish quantization’s effects on various aspects of the video, such as quality [59, 64], text-video alignment [17], and temporal consistency [32]. To overcome this, we propose to “decouple” these aspects by utilizing corresponding evaluation metrics, and introduce “metric-decoupled” quantization sensitivity estimation method to assist in mixed precision bit-width allocation. We summarize our contributions as follows:

1. We conduct extensive analysis and identify challenges for DiT quantization and design a specialized quantization scheme, “ViDiT-Q”, to address these issues.
2. We identify the quantization “bottlenecked” in lower bit-widths and propose a metric-decoupled mixed-precision method to mitigate this problem.
3. We validate the effectiveness of “ViDiT-Q” on extensive DiT models for both image and video generation. It achieves W8A8 with negligible metrics loss, and “ViDiT-Q-MP”

achieves W4A8 with minimal visual degradation, resulting in 2-2.5x memory saving and 1.5x latency speedup.

## 2 Related Works

### 2.1 Diffusion Transformers for Image and Video Generation

Diffusion Transformers (DiTs), which employ Transformers [60] to replace the CNN-based diffusion backbones (U-Net [50]) in prior research [45, 49], have achieved remarkable performance in visual generation. **Image Generation:** DiT [44] and UViT [1] pioneer the use of transformers as diffusion backbones. Subsequent research [14, 36, 71] makes further advancements in architecture design and training methods. PixArt- $\alpha$  [6] explores text-to-image generation with DiTs, PixArt- $\Sigma$  [5] extends generation resolution to 4K. **Video Generation:** Early video generation models [13, 20, 21] mainly adopted CNN backbones. Latte [37] pioneer the use of transformers for text-to-video generation. The success of SORA [41] inspire the development of video diffusion transformers such as GenTron [7], and open-sora STDiT [22]. Both high-resolution image generation and multi-frame video generation add to hardware costs, **necessitating efficiency improvements.**

### 2.2 Image and Video Generation Evaluation Metrics

Visual generation can be evaluated from multiple aspects, and many metrics are introduced accordingly. **Image Metrics:** FID [18] and IS [51] are two commonly adopted metrics for measuring the Inception network feature difference between generated and reference images for quality and fidelity assessment. ClipScore [17] evaluates how well the generated image follows the prompt instruction (text-image alignment), while ImageReward [67], HPS [65] incorporates human preference by collecting actual user data to train the reward model. **Video Metrics:** FVD extends the feature-based metric FID to the video domain. CLIPSIM [63] estimates the similarity between video and text instructions. CLIP-temp [11] measures the semantic similarity between video frames. Flow-score is proposed as part of the video evaluation benchmark EvalCrafter [33] to assess motion quality. EvalCrafter also adopts DOVER [64] for video quality assessment. These **metrics from multiple aspects should be considered** when evaluating the effect of quantization.

### 2.3 Model Quantization

Post Training Quantization (PTQ) has proven to be an efficient and effective model compression method [38]. It converts the floating-point data into low-bit integers, the process could be represented as:  $x_q = \text{round}(\text{clamp}((x - z)/s, -2^{B-1}, 2^{B-1} - 1))$ . The  $s$  (scale) and  $z$  (zero point) are quantization parameters, which are determined offline based on a set of calibration data with  $S = \max(\text{abs}(x)); z = (\max + \min)/2$ . **Diffusion Model:** Prior research Q-Diffusion [27] and PTQD [16] apply post training quantization for diffusion models. Other research [4, 16, 31, 54, 56, 68] continue to improve diffusion quantization from the unique timestep dimension. **Transformer:** Prior research has made advances in quantizing transformers for both the ViT [23, 30, 35, 70] and language model quantization [9, 26, 66, 69]. To the best of our knowledge, **no prior research has investigated the DiT quantization.** We identify the challenges of applying existing methods to DiT quantization, and design a novel specialized technique as a solution.

## 3 Methods

The framework of the **ViDiT-Q** quantization method is illustrated in Fig. 3. We start by analyzing the unique challenges for quantizing DiTs (Sec. 3.1). Then, we introduce an improved quantization scheme, ViDiT-Q, specialized for DiTs to address these challenges (Sec. 3.2). Further, we develop a metric-decoupled mixed-precision method to improve quantization under lower bit-widths (Sec. 3.3).

### 3.1 Challenges for DiT Quantization

As presented in Sec. 1 and Fig. 2, directly applying existing diffusion quantization methods to DiTs leads to notable degradation for W8A8 and produces blank images for W4A8. We delve into the reasons for these failures by examining the data distributions. As illustrated in Fig. 4, we discover

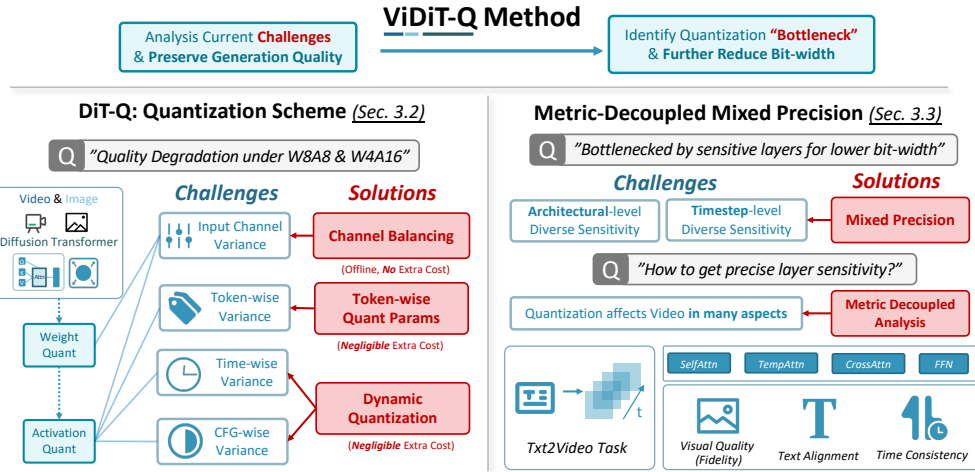


Figure 3: **The overall framework of ViDiT-Q.** We design quantization scheme tailored for DiT’s unique challenges, and introduce mixed precision specialized for video generation.

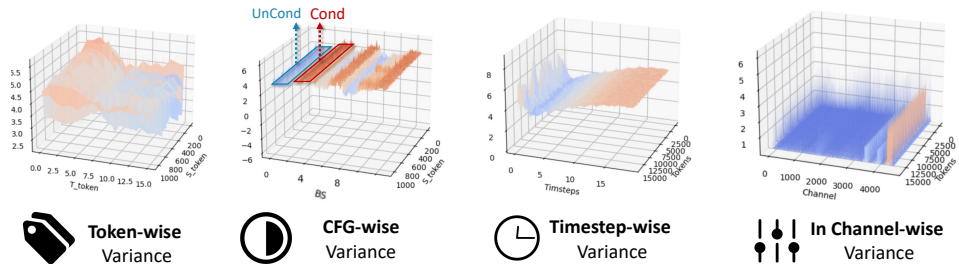


Figure 4: **The challenges of DiT quantization.** The key unique challenge of DiT quantization lies in the variance of data at multiple levels. The above figures depicts the distribution of activation along different dimensions for STDiT model’s layer “blocks.26.ffn.fc1”.

that the unique challenge for DiT quantization lies in the **data variance in multiple levels**. Existing methods use **fixed and coarse-grained quantization parameters**, which struggle to handle highly variant data. We summarize the data variance in **four levels** as follows:

- **Token-wise Variance:** In DiTs, the activation consists of visual tokens, We observe notable difference exists between these tokens.
- **CFG-wise Variance:** For conditional generation, the classifier-free-guidance [19] (CFG) conducts two separate forwards with and without the control signal (often implemented with batch). We observe notable difference in activation range between the conditional part (red square) and the unconditional part (blue square).
- **Timestep-wise Variance:** Diffusion method iterates the model for multiple timesteps. We observe notable variance in activation for the same layer across timesteps.
- **Input Channel-wise Variance:** For both the weight and activation, we witness significant difference across different input-channel.

### 3.2 ViDiT-Q: Improved DiT Quantization Scheme

#### 3.2.1 Token-wise Quantization

The primary distinction between DiT and previous UNet-based diffusion models lies in the feature extraction operator. While U-Net employs convolution layers for local pixel feature aggregation, DiT utilizes linear and attention layers for processing image tokens. In DiT quantization, we perform hardware profiling on the GPU (Ref the appendix A.4 for more details) and observe that, when combined with flash attention [8], linear layers account for over 77% of the latency and nearly 100% of the memory cost. Therefore, following prior literature on transformer-based Language Model (LM) quantization [3, 26, 66], we focus on quantizing linear layers.

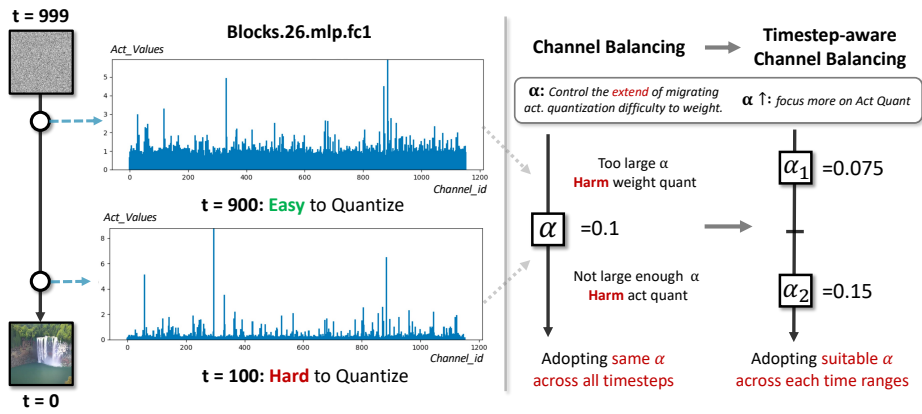


Figure 5: **The illustration of timestep-wise channel balancing.** Because the activation quantization difficulty changes across timesteps, we assign different  $\alpha$  for each range to adjust the difficulty.

In convolutions, neighboring pixel values are summed together. During quantization, these values need to share the same quantization parameters for practical acceleration [38]. Therefore, current CNN-based diffusion quantization methods primarily adopt tensor-wise quantization parameters. However, as discussed earlier, token values exhibit notable variation in DiT activation. Applying the same set of quantization parameters to them leads to large quantization errors. In contrast to convolution layers, linear layers do not require summation between tokens. Thus, **token-wise quantization parameter assignment** is applicable, which can mitigate “**token-wise variance**” challenge. We further analyze the extra cost of introducing token-wise quantization parameters. The token-wise quantization parameter size is negligible compared with the activation (around 1/1000).

### 3.2.2 Dynamic Quantization

The “CFG-wise variance” and “Timestep-wise variance” present two unique challenges for diffusion models. To the best of our knowledge, no prior research addresses the CFG-wise variation. Regarding timestep-wise variance, prior diffusion quantization methods [15, 54] tackle this issue by adopting timestep-wise quantization parameters through calibration or gradient-based optimization. However, such a process is costly and may encounter challenges when generalizing to unseen conditions.

Inspired by recent advances in language model quantization [66, 69], we adopt **dynamic quantization that calculates the quantization parameters online for activation quantization**. We utilize the simple minmax quantization scheme, so the cost of obtaining quantization parameters online is acceptable (obtaining data’s max and min). Profiling results (refer to the appendix A.4 for details) confirm that the additional latency is less than 1/100 of the layer inference. Moreover, the cost could be further mitigated with kernel fusion techniques [61]. While dynamic quantization has been utilized for language models, we emphasize its importance for diffusion models due to the unique “**CFG-wise variance**” and “**timestep-wise variance**” challenges.

### 3.2.3 Timestep-aware Channel Balancing

The above challenges exist in activation quantization only. However, for both weights and activations, there exists **input channel-wise variance**. A few channels have significantly larger values ( $> 100\times$ ) than others, leading to the majority of values being quantized to zero. Following prior channel balancing quantization methods [28, 66], we introduce a per-channel balancing mask  $s \in \mathcal{R}^{C_i}$ . By dividing the activation with  $s$  and multiplying  $s$  with weights, it shifts the quantization difficulty from activation to weights, and vice versa. The channel balancing mask could be calculated as follows:

$$Y = (X \text{diag}(s)^{-1}) \cdot ((\text{diag}(s)W)) = \hat{X} \cdot \hat{W}; \quad s_i = \max(|X_i|)^\alpha / \max(|W_i|)^{1-\alpha}, \quad (1)$$

where  $X, Y, W$  represents the input activation, output activation, and weights. The  $s$  is the aforementioned channel-wise balancing mask, and  $\alpha$  is a hyperparameter. Channel balancing can effectively alleviate the input channel-wise variance. However, we empirically discover that it is very sensitive to  $\alpha$  choices, and **still yields unsatisfactory performance**. We further investigate such limitations and observe that the activation distribution is notably different across timesteps. As shown in Fig. 5 for the blocks.27.ffn.fc2 layer, the outlier values are more prominent for later denoising steps ( $t=100$ ).

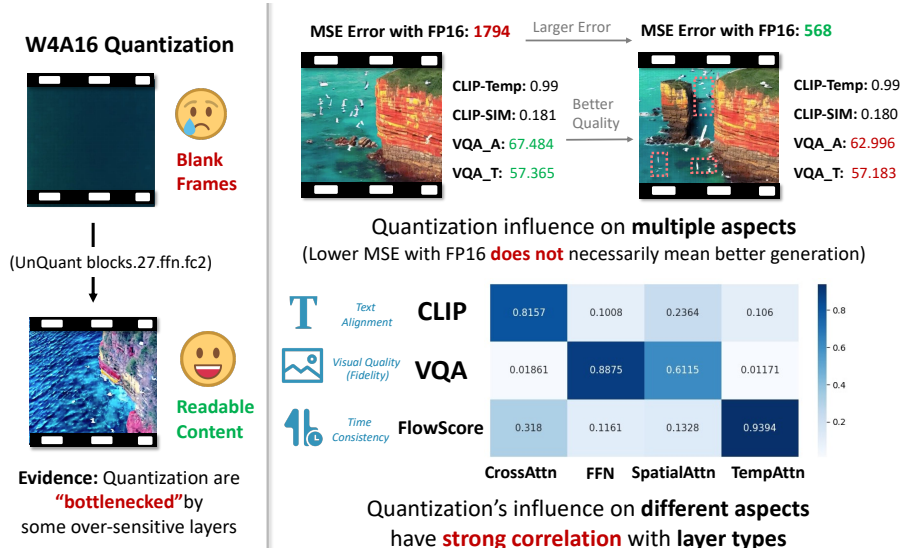


Figure 6: **The quantization “bottleneck” phenomenon and the motivation of metric decoupled analysis.** Some highly sensitive layers hinder the quantization, and quantization’s effect on multiple aspects needs decoupling for accurate sensitivity. The values in the lower-right heatmap are calculated by applying z-score normalization [43] to the metric changes when certain layer types are quantized.

**The “quantization difficulty” changes significantly across timesteps.** Therefore, employing the same alpha for earlier stages may shift too much difficulty from activations to weights, harming the weight quantization (especially for lower bit-width such as W4). For latter stages, the  $\alpha$  is not big enough, and the activation quantization is not sufficiently compensated.

To address this issue, as presented in Fig. 5, we design a **timestep-aware channel balancing** that assigns different  $\alpha$  for different timesteps. We empirically find that the distribution of most layers starts to become acute after halfway. To minimize the cost of  $\alpha$  tuning, we equally divide the diffusion process into two halves. Specifically, it could be inferred from the distribution that the former  $\alpha_1$  should be smaller than globally tuned  $\alpha$ , and vice versa for the latter  $\alpha_2$ . With the assistance of such prior, we can efficiently acquire a well-performing  $\alpha$  within 20 PTQ processes ( $< 1$  GPU hour).

### 3.3 Metric Decoupled Mixed Precision Design

#### 3.3.1 Arch-level and Timestep-level Mixed Precision

As seen in Fig. 2, for lower bit-width settings (e.g., W6A6 and W4A8), quality degradation is still observed. When investigating the reasons for such failure, we find that some layers, despite having relatively low quantization error, could significantly affect quantization. As presented in Fig. 6, W4A16 quantization produces blank images, but simply skipping quantizing the “blocks.27.ffn.fc2” layer in the STDiT model results in output containing readable content. This reflects that **quantization is “bottlenecked” by some highly sensitive layers**. To address such “bottleneck” phenomenon, an intuitive solution is to assign higher bit-widths to “protect” these sensitive layers. We identify variance in quantization sensitivity across the layers and timesteps, and propose applying mixed precision in these two levels.

#### 3.3.2 Metric Decoupled Sensitivity Analysis

The main challenge with mixed precision allocation lies in identifying the sensitive layers. Prior literature measures quantization sensitivity by quantizing specific layers alone and then calculating the Mean Squared Error (MSE) with the floating-point output to indicate their sensitivity. However, video evaluation has multiple aspects, **the MSE error alone may not accurately reflect overall generation performance**. As illustrated in the upper-right part of Fig. 6, although the left video has a higher MSE error, it exhibits better visual quality than the right video, which fails to correctly generate birds but shows some white fractures (denoted with red squares). This underscores the necessity to enhance sensitivity analysis metrics for video generation.

In exploring sensitivity analysis design, we find that **quantization’s impact on various aspects of generation quality is strongly correlated with layer types**. As observed in the lower-right part

Table 1: **Performance of ViDiT-Q text-to-video generation on VBench evaluation benchmark suite.** The bit-width “16” represents FP16 without quantization. The “ViDiT-Q-MP” denotes quantization with mixed precision, which keeps a small portion of layers as W8A8.

Model	Method	Bit-width (W/A)	Imaging Quality	Aesthetic Quality	Motion Smooth.	Dynamic Degree	BG. Consist.	Subject Consist.	Scene Consist.	Overall Consist.
	-	16/16	63.68	57.12	96.28	56.94	96.13	90.28	39.61	26.21
STDiT (OpenSORA)	Naive PTQ	8/8	56.52	53.64	95.85	69.44	93.54	85.30	29.28	25.42
	ViDiT-Q	8/8	63.48	56.95	96.14	61.11	95.84	90.24	38.22	26.06
	ViDiT-Q-MP	6/6	62.07	57.03	95.86	62.50	95.86	89.34	39.46	26.41
	ViDiT-Q-MP	4/8	61.07	55.37	95.69	58.33	95.23	88.72	36.19	25.94

Table 2: **Performance of text-to-video generation on UCF-101 Dataset.** The description of metrics is provided in Sec. 4.1, unless specified with  $\downarrow$ , higher metric values denote better performance.

Model	Method	Bit-width (W/A)	FVD( $\downarrow$ )	FVD-FP16( $\downarrow$ )	CLIPSIM	CLIP-T	VQA-Aesthetic	VQA-Technical	Flow Score.	Temp. Flick.
	-	16/16	136.87	0.00	0.1996	0.9978	41.63	56.64	2.24	97.53
STDiT	Naive PTQ	8/8	154.92	50.72	0.1993	0.9968	27.52	35.50	2.61	97.02
	ViDiT-Q	8/8	141.13	15.52	0.1995	0.9978	43.59	55.36	2.32	97.45
	Naive PTQ	4/8	544.34	637.02	0.1868	0.9982	0.16	0.13	1.61	99.90
	ViDiT-Q-MP	4/8	129.10	60.13	0.1995	0.9977	33.98	47.65	1.89	97.57
Latte	-	16/16	99.90	0.00	0.1970	0.9963	36.33	91.23	3.37	96.22
	Naive PTQ	8/8	98.75	73.82	0.1981	0.9950	27.62	50.52	3.53	95.35
	ViDiT-Q	8/8	110.96	20.83	0.1959	0.9962	30.26	80.32	3.14	95.95
	Naive PTQ	4/8	183.52	239.08	0.1719	0.9929	5.62	0.41	66.06	65.14
	ViDiT-Q	4/8	95.04	79.11	0.1943	0.9971	21.76	32.17	2.84	95.57

of Fig. 6, cross-attention layers contribute significantly more than other layers to “content change”. Similarly, “visual quality” is mainly affected by spatial attention and FFN layers, while “temporal consistency” is primarily influenced by temporal attention layers. Building upon this finding, we propose to “**disentangle**” the mixed influences of quantization on multiple aspects of generation performance and develop a metric-decoupled sensitivity analysis method. We categorize layers into three groups based on their effects on different aspects and use the corresponding metrics as sensitivity measures. Subsequently, we identify layers with significantly higher relative sensitivity and adjust their bit-width accordingly. For timestep mixed precision, we divide the entire process into four equal ranges and quantize them separately to identify the most sensitive part. Following these principles, we design mixed-precision versions of ViDiT-Q for W6A6 and W4A8.

## 4 Experiments

### 4.1 Implementation Details and Experimental Settings

We evaluate the performance of ViDiT-Q on a variety of models, bit-widths and evaluation settings. For more detailed settings, please refer to Appendix B and C.

**Quantization Scheme:** We adopt the simple minmax quantization scheme. The quantization parameters for activation are dynamic and computed online with negligible overhead. The channel balancing  $\alpha$  and the mixed precision plan are determined offline based on the calibration data.

**Mixed Precision Strategy** For W6A6, we discover the FFN layers for blocks 6 and 26 are significantly more sensitive, since they account for  $< 1\%$  of the overall latency, we maintain them as FP16, and keep the rest layers as W6A6. For W4A8, we witness notable higher sensitivity for most FFNs, for simplicity, we set all FFN layers (15% layers) as W8A8, and set the rest as W4A8. Also, we discover the first quarter of timesteps is more sensitive, and set them to W8A8.

**Video Genration Evaluation Settings:** We apply ViDiT-Q to Latte [37] and STDiT (open-sora) [22] and evaluate them in two settings. (1) **Benchmark suite:** We evaluate the quantized model on VBench [24] to provide comprehensive results. Following prior research [48], we select 8 major dimensions from VBench. (2) **Multi-aspects metrics:** For more flexible evaluation, we select representative metrics for each aspect, and measure them on UCF-101 [55] and open-sora [22]

Table 3: **Comparison of quantization methods on open-sora prompt set.** Due to the lack of existing DiT quantization method, we adapt baseline quantization methods to DiT.

Model	Method	Bit-width (W/A)	CLIPSIM	CLIP-Temp	VQA-Aesthetic	VQA-Technical	Flow Score.
STDiT	-	16/16	0.1797	0.9988	63.4014	50.4597	0.9751
	Naive PTQ	8/8	0.1956	0.9988	48.2358	25.5961	0.6081
	PTQ4DM [52]	8/8	0.1812	0.9984	50.0674	25.1344	0.6335
	Q-Diffusion [27]	8/8	0.1781	0.9987	51.6834	38.2680	0.6473
	ZeroQuant [69]	8/8	0.1938	0.9988	58.3127	51.5312	0.6442
	ViDiT-Q	8/8	0.1950	0.9991	60.7025	54.6361	0.8865
	Naive PTQ	6/6	0.1912	0.9964	14.6374	10.6919	33.611
	PTQ4DM [52]	6/6	0.1804	0.9977	20.9610	8.6566	0.7600
	ViDiT-Q-MP	6/6	0.1794	0.9983	60.8124	53.4413	1.2266
	Naive PTQ	4/8	0.2010	0.9986	0.1765	0.0863	1.5722
	PTQ4DM [52]	4/8	0.1727	0.9981	0.4781	0.2672	1.5298
	SmoothQuant [66]	4/8	0.1910	0.9989	31.9626	22.8467	0.5594
	ViDiT-Q-MP	4/8	0.1809	0.9989	60.6158	49.3838	1.1278

Model	Method	Bit-width (W/A)	FID(↓)	CLIP(↑)	IR(↑)
Pixart- $\alpha$	-	16/16	73.338	0.258	0.901
	Naive	8/8	115.14	0.226	-0.953
		4/8	108.40	0.234	-0.725
	ViDiT-Q	8/8	75.613	0.259	0.917
		4/8	74.326	0.257	0.887
Pixart- $\Sigma$	-	16/16	72.699	0.262	0.929
	Naive	8/8	302.12	0.163	-2.236
		4/8	287.80	0.165	-2.219
	ViDiT-Q	8/8	72.845	0.263	0.926
		4/8	71.936	0.264	0.944

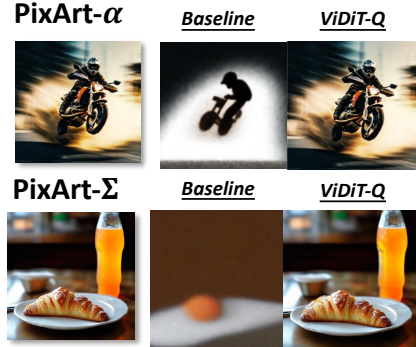


Figure 7: **Performance of ViDiT-Q text-to-image generation on COCO.** Left: The metric scores of PixArt- $\alpha$  and PixArt- $\Sigma$  quantization. Right: Generated image comparison of W8A8 quantization.

prompt sets. Following EvalCrafter [33], we select *CLIPSIM* and *CLIP-Temp* to measure the text-video alignment and temporal semantic consistency, and DOVER [64]’s video quality assessment (*VQA*) metrics to evaluate the generation quality from aesthetic (high-level) and technical (low-level) perspectives, *Flow-score* and *Temporal Flickering* are used for evaluating the temporal consistency. The commonly adopted *FVD* [59] is also provided. Specifically, due to the lack of ground-truth videos for prompt-only datasets, inspired by [57], we also report *FVD-FP16* which chooses the FP16 generated video as ground-truth. The above metrics are evaluated on 101 prompts (1 for each class) for UCF-101, and 10 example prompts for open-sora. We adopt the class-conditioned Latte model trained on UCF-101. We use the 20-steps DDIM solver with CFG scale of 7.0 for Latte, and 100-steps DDIM with CFG scale of 4.0 for STDiT.

**Image Evaluation Settings:** We apply ViDiT-Q to PixArt- $\alpha$  and PixArt- $\Sigma$  model on the the first 1024 prompts of COCO annotations [29]. We choose *FID* [18] for fidelity evaluation, *Clipscore* [17] for text-image alignment, and *ImageReward* [67] for human preference in multiple aspects. Following the original code implementation, we use the 20-steps DPM-solver with CFG scale of 4.5.

## 4.2 Main Results

We briefly conclude key findings here, for more detailed analysis, please refer to Appendix Sec. C.

**Text-to-video generation on VBench and UCF-101:** As presented in Tab. 1 and Tab. 2, while naive W6A6 and W4A8 quantization generates pure noise or blank images (Ref Fig. 2 and Appendix for visual example), ViDiT-Q consistently preserves generation quality for all bit-widths. ViDiT-Q’s W4A8 even outperforms baseline W8A8. It’s worth noting that some abnormal scores are due to the baseline’s failure to generate meaningful contents. Besides, having a flow-score that is either too high



Table 4: **Ablation studies of ViDiT-Q techniques..** By gradually incorporating ViDiT-Q’s techniques, W4A8 quantization improves from failure to acceptable.

Methods			Bit-width	CLIPSIM	CLIP-Temp	VQA-Aesthetic	VQA-Technical	Flow Score.
Dynamic	Channel Balance	Timestep-aware CB	(W/A)					
-	-	-	16/16	0.180	0.998	64.198	51.904	1.427
-	-	-	4/8	0.201	0.997	0.178	0.086	1.572
✓	-	-	4/8	0.196	0.998	32.217	10.994	0.884
✓	✓	-	4/8	0.191	0.999	31.963	22.847	1.117
✓	✓	✓	4/8	0.199	0.999	51.823	23.215	1.189

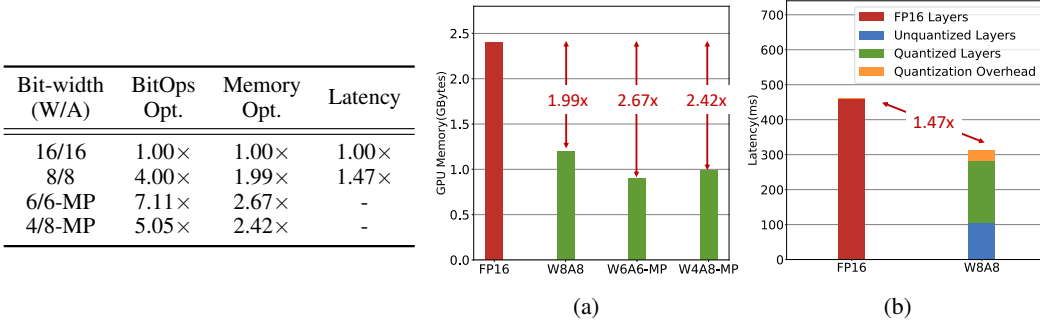


Figure 8: **The illustration of ViDiT-Q’s hardware resource savings.** Left: The comparison of efficiency and performance under different ViDiT-Q quantization configurations. The “6/6-MP” and “4/8-MP”’s latency could not be measured for now due to lack of GPU kernels (Ref appendix A.4 for details). Right: (a) ViDiT-Q’s memory optimization, (b) ViDiT-Q’s latency breakdown under W8A8.

or too low than FP videos indicates degradation. Also, we discover that the FVD fluctuates heavily with less data (101 videos). We find that FVD-FP16 shows better correlation with visual perception.

**Comparison with other quantization methods:** We compare ViDiT-Q with other quantization methods in Tab. 3. Due to the lack of a baseline DiT quantization method, we adapt prior diffusion and transformer quantization methods to DiTs. For W8A8, ViDiT-Q outperforms other quantization methods and achieves similar metric scores to FP16. For the challenging W4A8, only SmoothQuant [66] could prevent generation failure. ViDiT-Q with mixed precision further improves performance to be comparable with baseline W8A8 quantization.

**Text-to-image generation on COCO:** As presented in Fig. 7, ViDiT-Q significantly improves generation quality from blurred and unreadable to almost identical to the FP generation. The metric scores are similar and even higher than FP16 baseline for W8A8 and W4A8. More analysis and generated images are presented in Appendix appendix C.3.

**Ablation Studies:** We conduct ablation studies by gradually incorporating ViDiT-Q’s techniques to the challenging W4A8 quantization. As shown in Tab. 4, the introduction of dynamic quantization marks the beginning of the generation meaningful content. The subsequent integration of channel balancing, along with its timestep-wise version, further improves generation quality. Nevertheless, we still observe significant degradation, underscoring the necessity for mixed precision strategies. Detailed analysis and video examples are presented in Appendix appendix C.4.

### 4.3 Hardware Resource Savings

**Memory footprint reduction.** Fig. 8 (a) shows the GPU memory usage of ViDiT-Q and the FP16 baseline. ViDiT-Q can reduce the memory from two aspects: (1) Weight quantization reduces the allocated memory for storing model weights. (2) Activation quantization reduces allocated memory to store intermediate activations. Combining the two benefits, ViDiT-Q can effectively reduce the peak memory footprint by 2×, 2.42× and 2.67× under the W8A8, W4A8-MP and W6A6-MP.

**Latency speedup.** We present the latency speedup in Fig. 8-(b). Due to lack of W6A6/W4A8 or dynamic quantization kernels, we adopt the INT8 GPU kernels to estimate the latency speedup. Replacing FP16 layers with INT8 achieves ~ 2× acceleration. Considering the unquantizable layers (norms, nonlinears, attention), and the FP to INT conversion cost, the overall speedup is 1.47×. Detailed description of hardware resource measurements are provided in appendix A.4.

## **5 Conclusion and Limitations**

We design ViDiT-Q, a quantization method tailored for DiTs. With the assistance of metric decoupled mixed precision, ViDiT-Q achieves W4A8 quantization with minimum visual degradation. Despite achieving good performance, it still faces the following limitations: the timestep-wise mixed precision requires additional system-level optimization, the efficiency and precision of quantization sensitivity analysis still needs improvement, lower activation bit-width is essential for fully utilizing the acceleration potential of 4-bit weight. We aim to address these issues and further improve ViDiT-Q.

## **6 Acknowledgements**

This work was supported by National Natural Science Foundation of China (No. 62325405, 62104128, U19B2019, U21B2031, 61832007, 62204164), Tsinghua EE Xilinx AI Research Fund, and Beijing National Research Center for Information Science and Technology (BNRist). We thank for all the support from Infinigence-AI.

## References

- [1] Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *CVPR*, 2023.
- [2] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023.
- [3] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13169–13178, 2020.
- [4] Hanwen Chang, Haihao Shen, Yiyang Cai, Xinyu Ye, Zhenzhong Xu, Wenhua Cheng, Kaokao Lv, Weiwei Zhang, Yintong Lu, and Heng Guo. Effective quantization for diffusion models on cpus. *arXiv preprint arXiv:2311.16133*, 2023.
- [5] Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart- $\sigma$ : Weak-to-strong training of diffusion transformer for 4k text-to-image generation, 2024.
- [6] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart- $\alpha$ : Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023.
- [7] Shoufa Chen, Mengmeng Xu, Jiawei Ren, Yuren Cong, Sen He, Yanping Xie, Animesh Sinha, Ping Luo, Tao Xiang, and Juan-Manuel Perez-Rua. Gentron: Delving deep into diffusion transformers for image and video generation. *arXiv preprint arXiv:2312.04557*, 2023.
- [8] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.
- [9] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale, 2022. *CoRR abs/2208.07339*.
- [10] Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Hawq: Hessian aware quantization of neural networks with mixed-precision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 293–302, 2019.
- [11] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7346–7356, 2023.
- [12] Zhao et. al. Mixdq: Memory-efficient few-step text-to-image diffusion models with metric-decoupled mixed precision quantization. <https://a-suo Zhang.github.io/>, 2024. Accessed: 2024-05-20.
- [13] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. AnimateDiff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023.
- [14] Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. Diffit: Diffusion vision transformers for image generation. *arXiv preprint arXiv:2312.02139*, 2023.
- [15] 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.
- [16] 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.

- [17] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Bras, and Choi Yejin. Clipscore: A reference-free evaluation metric for image captioning. pages 7514–7528, 01 2021.
- [18] 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. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, page 6629–6640, Red Hook, NY, USA, 2017. Curran Associates Inc.
- [19] Jonathan Ho. Classifier-free diffusion guidance. *ArXiv*, abs/2207.12598, 2022.
- [20] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.
- [21] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.
- [22] HPC-AI Advisory Council. Open-Sora. <https://github.com/hpcaitech/Open-Sora>, 2024.
- [23] Xijie Huang, Zhiqiang Shen, and Kwang-Ting Cheng. Variation-aware vision transformer quantization. *arXiv preprint arXiv:2307.00331*, 2023.
- [24] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. *arXiv preprint arXiv:2311.17982*, 2023.
- [25] 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.
- [26] Shiyao Li, Xuefei Ning, Luning Wang, Tengxuan Liu, Xiangsheng Shi, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. Evaluating quantized large language models, 2024.
- [27] 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.
- [28] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv preprint arXiv:2306.00978*, 2023.
- [29] 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, 2015.
- [30] Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou. Fq-vit: Post-training quantization for fully quantized vision transformer. *arXiv preprint arXiv:2111.13824*, 2021.
- [31] Xuewen Liu, Zhikai Li, Junrui Xiao, and Qingyi Gu. Enhanced distribution alignment for post-training quantization of diffusion models. *arXiv preprint arXiv:2401.04585*, 2024.
- [32] Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tiejong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models. *arXiv preprint arXiv:2310.11440*, 2023.
- [33] Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tiejong Zeng, Raymond Chan, and Ying Shan. Evalcrafter: Benchmarking and evaluating large video generation models. 2023.

- [34] Zechun Liu, Zhiqiang Shen, Marios Savvides, and Kwang-Ting Cheng. Reactnet: Towards precise binary neural network with generalized activation functions. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 143–159. Springer, 2020.
- [35] Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems*, 34:28092–28103, 2021.
- [36] Zeyu Lu, Zidong Wang, Di Huang, Chengyue Wu, Xihui Liu, Wanli Ouyang, and Lei Bai. Fit: Flexible vision transformer for diffusion model, 2024.
- [37] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024.
- [38] 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.
- [39] NVIDIA. Nsight Systems. <https://docs.nvidia.com/nsight-systems/index.html>, 2024.
- [40] Nvidia. Nvidia cutlass release v3.4, 2024.
- [41] OpenAI. Video generation models as world simulators. <https://openai.com/index/video-generation-models-as-world-simulators/>, Year of Access.
- [42] Nilesh Prasad Pandey, Markus Nagel, Mart van Baalen, Yin Huang, Chirag Patel, and Tijmen Blankevoort. A practical mixed precision algorithm for post-training quantization. *arXiv preprint arXiv:2302.05397*, 2023.
- [43] GOPAL Patro and Kishore Kumar Sahu. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*, 2015.
- [44] William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023.
- [45] Dustin Podell, Zion English, Kyle Lacey, A. Blattmann, Tim Dockhorn, Jonas Muller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *ArXiv*, abs/2307.01952, 2023.
- [46] PyTorch. PyTorch Memory Management, 2023.
- [47] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.
- [48] Weiming Ren, Harry Yang, Ge Zhang, Cong Wei, Xinrun Du, Stephen Huang, and Wenhui Chen. Consisti2v: Enhancing visual consistency for image-to-video generation. *arXiv preprint arXiv:2402.04324*, 2024.
- [49] 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*, pages 10684–10695, 2022.
- [50] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.
- [51] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. NIPS’16, page 2234–2242, Red Hook, NY, USA, 2016. Curran Associates Inc.

- [52] 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.
- [53] Uriel Singer, Adam Polyak, Thomas Hayes, Xiaoyue Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. *ArXiv*, abs/2209.14792, 2022.
- [54] 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.
- [55] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.
- [56] Haojun Sun, Chen Tang, Zhi Wang, Yuan Meng, Xinzhu Ma, Wenwu Zhu, et al. Tmpq-dm: Joint timestep reduction and quantization precision selection for efficient diffusion models. *arXiv preprint arXiv:2404.09532*, 2024.
- [57] Siao Tang, Xin Wang, Hong Chen, Chaoyu Guan, Zewen Wu, Yansong Tang, and Wenwu Zhu. Post-training quantization with progressive calibration and activation relaxing for text-to-image diffusion models. *arXiv preprint arXiv:2311.06322*, 2023.
- [58] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020.
- [59] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. 2019.
- [60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [61] 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.
- [62] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8612–8620, 2019.
- [63] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. Godiva: Generating open-domain videos from natural descriptions. *arXiv preprint arXiv:2104.14806*, 2021.
- [64] Haoning Wu, Erli Zhang, Liang Liao, Chaofeng Chen, Jingwen Hou, Annan Wang, Wenxiu Sun, Qiong Yan, and Weisi Lin. Exploring video quality assessment on user generated contents from aesthetic and technical perspectives. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20144–20154, 2023.
- [65] 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.
- [66] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models, 2024.
- [67] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation, 2023.

- [68] Yuewei Yang, Xiaoliang Dai, Jialiang Wang, Peizhao Zhang, and Hongbo Zhang. Efficient quantization strategies for latent diffusion models. *arXiv preprint arXiv:2312.05431*, 2023.
- [69] 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.
- [70] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *European conference on computer vision*, pages 191–207. Springer, 2022.
- [71] Hongkai Zheng, Weili Nie, Arash Vahdat, and Anima Anandkumar. Fast training of diffusion models with masked transformers, 2024.

## A Hardware Experiments Details

### A.1 Profiling Settings

We evaluate the latency and memory usage of ViDiT-Q on the Nvidia RTX 4080 GPU using CUDA 12.1. All profiling is conducted with a batch size of 1. Since there are no open-source kernels supporting dynamic W8A8 quantization on GPU, we demonstrate in appendix A.3 that the additional cost of dynamic quantization over static quantization is negligible. For our analysis, we utilize static quantization GPU kernels implemented based on the Cutlass library [40] for latency and memory measurement. Memory usage is estimated using PyTorch Memory Management APIs [46], while inference latency is estimated with NVIDIA Nsight tools [39].

### A.2 Motivation for quantizing linear layers only

In Sec. 3, we mention that we focus on quantizing the linear layers and leave the attention computation unquantized. We elaborate on the reason for this focus here. In Fig. 9, we visualize the detailed latency breakdown for an STDiT model block. The 'attention computation' includes the matrix multiplication for query and key embedding to generate the attention map, and the multiplication of the attention map with the value embedding. The QKV linear mapping and the projection after attention aggregation are not included, as these are linear layers that can be quantized. As shown, when utilizing FlashAttention, the latency cost of attention computation accounts for only 14.3% of the overall latency. Additionally, FlashAttention minimizes the activation memory usage for storing the attention map. Therefore, we focus on the primary cost: the linear layers. We quantize all linear layers except for the "t embedding", "y embedding" and "final layer", they appear at the start or end of the model, and have smaller channel sizes. They account for only negligible amount of computation ( $< 1/1000$  overall latency), therefore we maintain them as FP16.

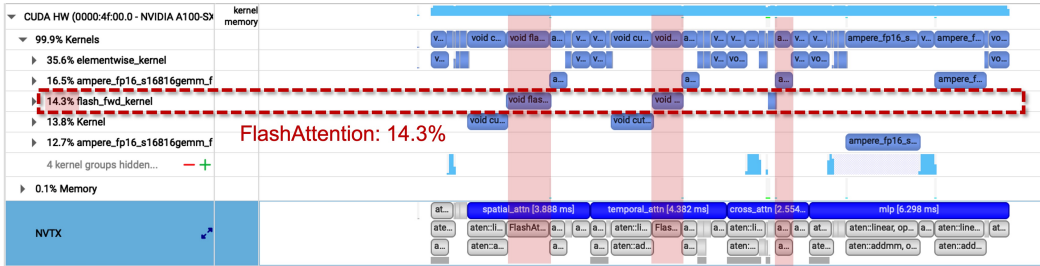


Figure 9: **The latency comparison of linear layers and attention computation.** When utilizing FlashAttention, the attention computation only takes up a small portion (14.3%) of the latency.

### A.3 Cost of ViDiT-Q quantization

**Dynamic quantization’s extra cost** As discussed in Sec. 3.2.2, dynamically calculating the quantization parameters online is vital for preserving the performance of diffusion transformers. ViDiT-Q adopts the easy-to-implement minmax quantization scheme. In this case, the extra cost of determining the quantization parameter online is obtaining the data’s min and max, which is negligible compared with the linear layer computation. To verify this, we conduct profiling with NVIDIA Nsight System. As seen from Fig. 10, the latency of the linear layer takes  $\sim 10$  ms, while the minmax operator (“reduce\_kernel”) only costs  $\sim 100\mu\text{s}$  (around  $1/100\times$ ). Furthermore, the extra cost of obtaining the minmax could be further reduced with the kernel-fusion technique [61]. The minmax operator is an element-wise reduced operation, which is similar to the “normalization” and “nonlinear activation” layers, which often occurs before the linear layer. By fusing the minmax operator into the previous layer, the overhead could be significantly reduced. To sum up, the extra cost of dynamic quantization over static quantization is negligible.

**Channel balancing’s cost:** As discussed in Sec. 3.2.3, the channel balancing methods introduce an additional channel-wise mask  $s$  to balance the ‘quantization difficulty’ between weight and activation. Prior literature [28, 66] has shown that the extra cost of multiplying  $s$  with weights and dividing the activation by  $s$  can be eliminated by merging  $s$  into the prior layer’s computation. Most linear layers



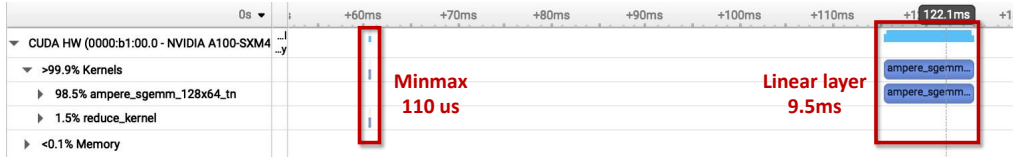


Figure 10: **The latency comparison of minmax operator and linear layer.** The minmax operator takes 1/100 of the linear layer’s latency.

are preceded by normalization layers (LayerNorm), which contain an affine transform. By dividing the weights in this affine transform, we achieve the same effect as dividing the linear layer’s activation by  $s$ . The multiplication of weights by  $s$  can be performed offline. This approach eliminates the extra cost of channel balancing.

**Post training quantization process’s cost** ViDiT-Q adopts a simple min-max quantization scheme, and the activation’s quantization parameters are computed online. Timestep-wise channel balancing requires several model inferences for  $\alpha$  tuning. As discussed in Sec. 3.2.3, guided by distribution analysis, proper  $\alpha$  values can be achieved with 10-30 PTQ inferences, taking less than 1 GPU hour. In contrast, some baseline quantization methods, such as Q-Diffusion, require gradient-based tuning of quantization parameters, which is costly and takes tens of GPU hours. For mixed precision, the major cost is determining sensitivity. Current ViDiT-Q’s sensitivity analysis requires iterative model testing, taking about 10 GPU hours. Enhancing the accuracy and speed of sensitivity analysis remains a future direction.

#### A.4 Estimation of Hardware Resource Savings

In Sec. 4.3, we present the efficiency improvement of ViDiT-Q and provide a more detailed analysis here. The efficiency improvement is shown in the table within Fig. 8 in the main paper. Due to the lack of W6A6 and W4A8 GPU kernels, we report the computational savings in BitOps, following prior literature [34]. Their latency is represented by W8A8’s latency. With negligible visual degradation under W6A6-MP and W4A8-MP, ViDiT-Q achieves  $7.11\times$  and  $5.05\times$  computation resource savings, respectively.

**Memory Optimization:** GPU memory usage comprises two parts: allocated memory for storing model weights and intermediate activations. With FlashAttention, the need for storing the huge attention map is eliminated. We find that the current peak memory for activation storage is the QKV embeddings. ViDiT-Q’s weight quantization reduces the model size, and activation quantization effectively reduces the size of QKV embeddings. By combining these benefits, ViDiT-Q effectively reduces the peak memory of the model. Both the “W6A6-MP” and “W4A8-MP” models adopt mixed precision and set some highly sensitive layers with higher bit-widths. Therefore, the memory optimization achieved is less than the theoretical statistics (W4A8-MP:  $2.42\times < W4A8: 2.67\times$ ).

**Latency Optimization:** As shown in Fig. 8-(b) in the main paper, we measure the latency of the three parts of the quantized model: the unquantized layers, the quantized layers, and the quantization overhead of FP16-to-INT8 conversion. These are summed to produce the overall latency. ViDiT-Q achieves a  $1.47\times$  latency speedup compared to the FP16 baseline. Further improvements in latency speedup with lower bit-width or system-level optimizations (e.g., W4A8 GPU kernels) remain for future exploration.

## B Detailed Description of Evaluation Metrics

### B.1 Benchmark Suite

Following VBench [24], our benchmark suite encompasses three key dimensions.

(1) **Frame-wise Quality** assesses the quality of each individual frame without taking temporal quality into concern.

- **Aesthetic Quality** evaluates the artistic and beauty value perceived by humans towards each video frame.
- **Imaging Quality** assesses distortion (e.g., over-exposure, noise) presented in the generated frames

(2) **Temporal Quality** assesses the cross-frame temporal consistency and dynamics.

- **Subject Consistency** assesses whether appearance of subjects in the video remain consistent throughout the whole video.
- **Background Consistency** evaluates the temporal consistency of the background scenes.
- **Motion Smoothness** evaluates whether the motion in the generated video is smooth and follows the physical law of the real world.
- **Dynamic Degree** evaluates the degree of dynamics by calculating average optical flow on each video frame.

(3) **Semantics** evaluates the video’s adherence to the text prompt given by the user. consistency.

- **Scene Consistency** evaluates whether the video is consistent with the intended scene described by the text prompt.
- **Overall Consistency** reflects both semantics and style consistency of the video.

We utilize three prompt sets provided by official github repository of VBench. We generate one video for each prompt for evaluation.

- **subject\_consistency.txt**: include 72 prompts, used to evaluate subject consistency, dynamic degree and motion smoothness.
- **overall\_consistency.txt**: include 93 prompts, used to evaluate overall consistency, aesthetic quality and imaging quality.
- **scene.txt**: include 86 prompts, used to evaluate scene and background consistency.

## B.2 Selected Metrics

**FVD and FVD-FP16**: FVD measures the similarity between the distributions of features extracted from real and generated videos. We employ one randomly selected video per label from the UCF-101 dataset (101 videos in total) as the reference ground-truth videos for FVD evaluation. We follow [2] to use a pretrained I3D model to extract features from the videos. Lower FVD scores indicate higher quality and more realistic video generation. However, due to relatively smaller video size (e.g. 101 videos in our case), employing FVD to evaluate video generation models faces several limitations. Small sample size cannot adequately represent either the diversity of the entire dataset or the complexity and nuances of video generation, leading to inaccurate and unstable results. To mitigate limitations above, we propose an enhanced metric, FVD-FP16, for assessing the semantic loss in videos generated by quantized models relative to those produced by pre-quantized models. Specifically, we utilize 101 videos generated by the FP16 model as ground-truth reference videos. The FVD-FP16 has significantly higher correlation with human perception.

**CLIPSIM and CLIP-temp**: The CLIPSIM and CLIP-temp metrics are computed using implementation from EvalCrafter [33]. For CLIPSIM, We use the CLIP-VIT-B/32 model [47] to compute the image-text CLIP similarity for all frames in the generated videos and report the averaged results. The metric quantify the discrepancy between input text prompts and generated videos. For CLIP-temp, we use the same model to compute the CLIP similarity of each two consecutive frames of the generated videos and then get the averages on each two frames. The metric indicates semantics consistency of generated videos.

**DOVER’s VQA**: We employ the Dover [64] method to assess generated video quality in terms of aesthetics and technicality. The technical rating(VQA-T) measures common distortions like noise, blur and over-exposure. The aesthetic rating(VQA-A) reflects aesthetic aspects such as the layout, the richness and harmony of colors, the photo-realism, naturalness, and artistic quality of the frames.

**Flow Score:** We employ flow score proposed by [33] to measure the general motion information of the video. we use RAFT [58], to extract the dense flows of the video in every two frames. Then, we calculate the average flow on these frames to obtain the average flow score of each generated video.

**Temporal Flickering:** We utilize the temporal flickering score provided by VBench [24] to measure temporal consistency at local and high-frequency details of generated videos. We calculate the average MAE(mean absolute difference) value between each frame.

## C Detailed Analysis of Experimental Results

In this section, we present more detailed analysis of the experimental results in Sec. 4.

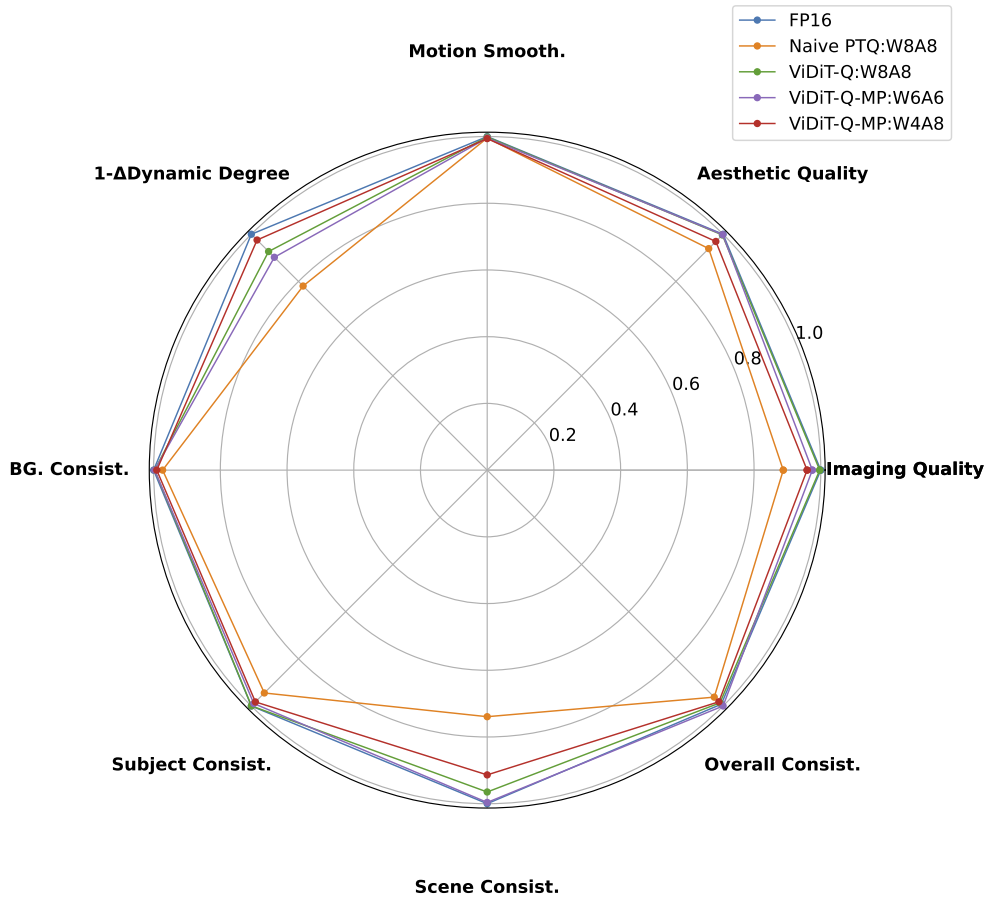


Figure 11: **The radar chart corresponding to the data presented in Table 1 from Sec. 4.1.** ViDiT-Q has a superior performance on VBench compared with the naive PTQ.

### C.1 Text-to-Video Performance on VBench

VBench is a comprehensive benchmark suite for video generation models, covering a wide range of dimensions, such as motion smoothness and subject consistency. The metric values of ViDiT-Q’s performance on VBench is presented in Tab. 1 Sec. 4. We visualize the Radar plot of the VBench performance in Fig. 11, the metric values are normalized by the maximum value in each dimension.

It’s clearly illustrated that ViDiT-Q achieves similar performance with FP16 for all bit-widths (W8A8, W6A6 mixed precision, W4A8 mixed precision), outperforming the Naive PTQ W8A8. We further analyze the generated video’s performance from three aspects as follows:

**Dynamic Degree:** Dynamic degree indicates the range of motion in the video, higher dynamic degree denotes more dynamic movement in the video. Lower dynamic degree denotes that the video barely moves, resembling a static image. Normally, higher dynamic degree is favored. However, in the quantization scenario, we discover that quantization often causes the generated videos to jitter and tremble. It is not favorable but results in notable dynamic degree value increase. In our experimental setting, **too high or too low dynamic degree means degradation**. Therefore, in the radar plot, using FP16 generated videos as the ground-truth reference, we use the  $(f_Q - f_{FP})/f_{FP}$  to denote “relative dynamic degree changes from FP generated videos”, and use  $1 - (f_Q - f_{FP})/f_{FP}$  as dynamic degree scoring in the radar plot. As illustrated Fig. 11 dynamic degree dimension, Naive PTQ W8A8’s scoring ( $< 0.8$ ) is notably lower than ViDiT-Q results. The video examples in Fig. 12 supports this finding. In Fig. 12c, the naive PTQ W8A8 generated buildings have jittering and glitches, and changes significantly across frames (ref the supplementary for the video). In contrast, both the FP16 and ViDiT-Q W8A8 generated buildings moves acutely.

**Consistency:** The consistency denotes whether some object remains consistent (does not disappear, change significantly) across frames. VBench evaluates consistency from the subject, scene, background, and overall level. From the Radar plot, we witness ViDiT-Q also notably outperforms naive PTQ, especially in the “scene consistency” dimension ( $< 0.8$ ). As seen in the aforementioned video example in Fig. 12c, the buildings (act as the “scene”) changes significantly across frames. It violates the scene consistency and lead to lower scoring. Also, as presented in Fig. 13c, the generated bear’s ear does not exist in earlier frames, and suddenly appears. This also reflects the degradation of subject consistency.

**Quality:** VBench evaluates the quality from both the aesthetic (composition and color), and imaging quality (clarity, exposure) dimension. Fig. 14 shows the example of Naive PTQ W8A8’s quality degradation. The color notably turns blue, and the mountain on the left is blurred. Similar color shifting degradation is also witnessed in Fig. 13c.



(a) FP16

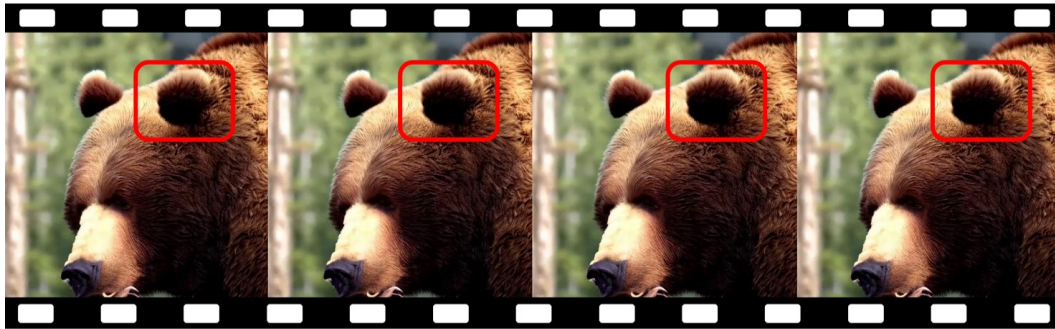


(b) ViDiT-Q: W8A8

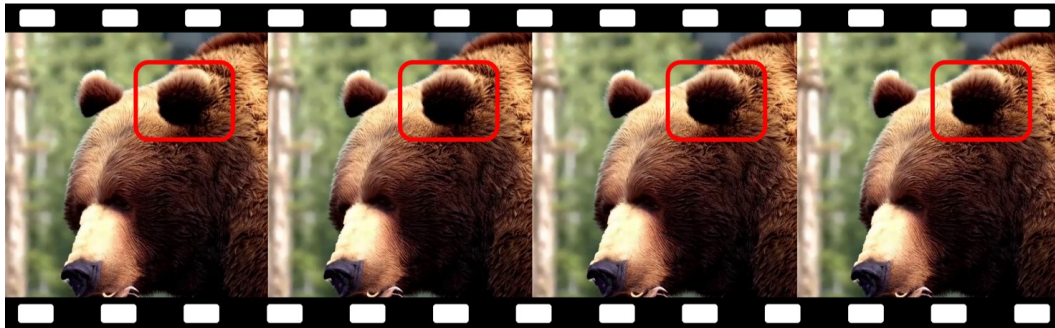


(c) Naive PTQ: W8A8

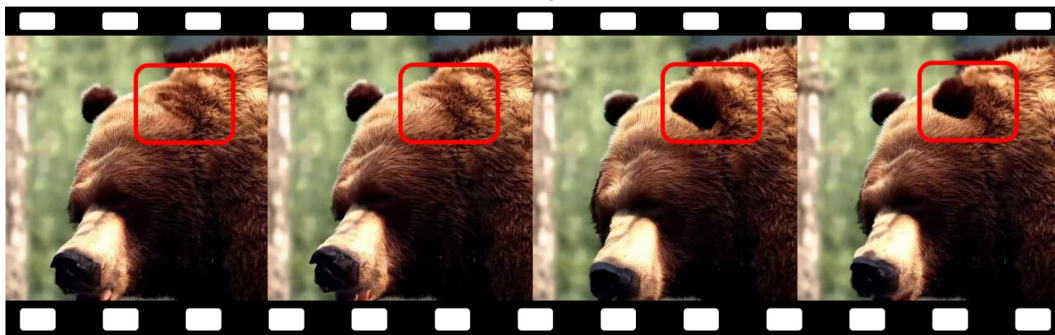
Figure 12: The qualitative results on VBench about the ViDiT-Q’s ability to maintain the dynamic degree.



(a) FP16



(b) ViDiT-Q: W8A8



(c) Naive PTQ: W8A8

Figure 13: The qualitative results on VBench about the ViDiT-Q's ability to maintain the consistency.



(a) FP16



(b) ViDiT-Q: W8A8



(c) Naive PTQ: W8A8

Figure 14: The qualitative results on VBench about the ViDiT-Q’s ability to maintain the image quality.

## C.2 Comparison with other quantization methods

In this section, we provide more detailed analysis of comparison with other quantization methods in Table. 3. We also visualize the radar plot in Fig. 15 of the statistics within the table. Similar to the abovementioned “dynamic degree” in VBench. The “flow Score” metric measures video dynamics. Therefore, we adopt similar  $1 - (f_Q - f_{FP})/f_{FP}$  for the flow score metric. For all bit-width. ViDiT-Q achieves similar performance with FP16, while baseline methods notably degrades or even fails. We will further elaborate the performance from each dimension with the assistance of example videos in Fig. 16.

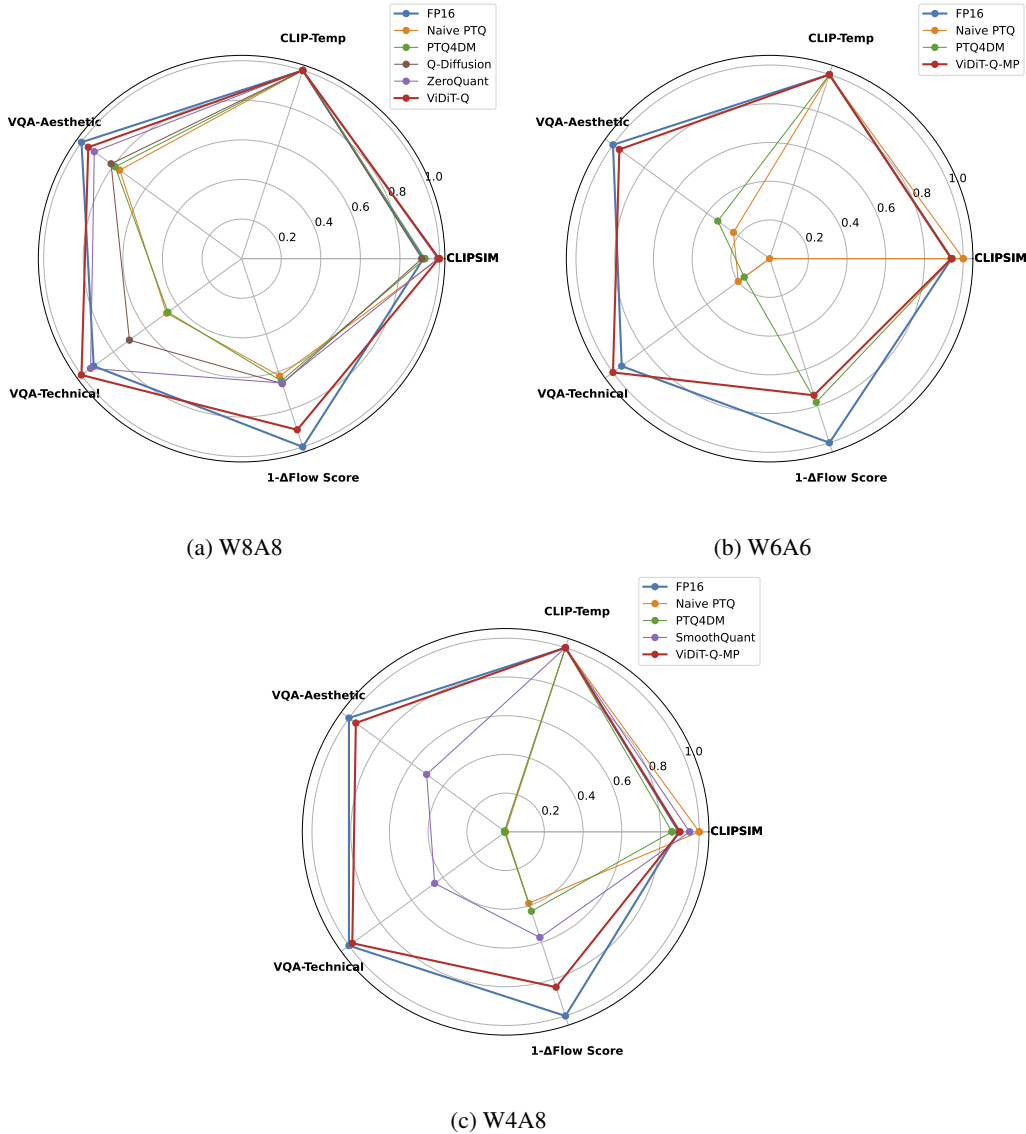


Figure 15: **The radar chart corresponding to the data presented in Table 3 from Sec. 4.2.** We compare ViDiT-Q with other quantization method designed for LLMs and UNet based diffusion models.

**CLIP-Temp and CLIPSIM:** CLIP-temp measures the similarity between frames. Despite baseline quantization methods faces subject and scene consistency degradation between frames, the main content of the frame does not change much. Therefore, we witness similar CLIPSIM score (close to 1.) for all methods. CLIPSIM measures the text-video alignment, which represents how well the



generated video follows the text instruction. Except for some extreme cases (Naive,PTQ4DM for W6A6,W8A8), which fail to generate reasonable contents, the CLIPSIM score remain similar.

**Flow Score:** Flow Score measures the dynamic degree of the video. We witness both higher and lower flow score for quantized model. Some videos witness flickering and glitches similar to Fig. 13c, and others have reduced motion (Refer to the videos in the supplementary for visual effect). The W6A6 Naive quantization generates pure noise, it significantly changes over frames and results in abnormally large Flow Score values.

**VQA:** The VQA metric we adopted evaluates video quality from two perspectives: the aesthetic and the technical (similar to VBench). As could be seen from Fig. 15, ViDiT-Q achieves VQA-A and VQA-T similar to FP16, while other baseline methods notably degrades. It's consistent with the visual perception from examples in Fig. 16. Naive PTQ could only generate readable content in W8A8, it generates pure noise for W6A6, and blank image for W4A8. PTQ4DM generates turtle that are hardly recognizable for W8A8 and W6A6, and also produce nearly blank images for W4A8. For improved quantization techniques Q-Diffusion and SmoothQuant, although the main object (turtle) is distinguishable, its movement is still unnatural and violates physics.

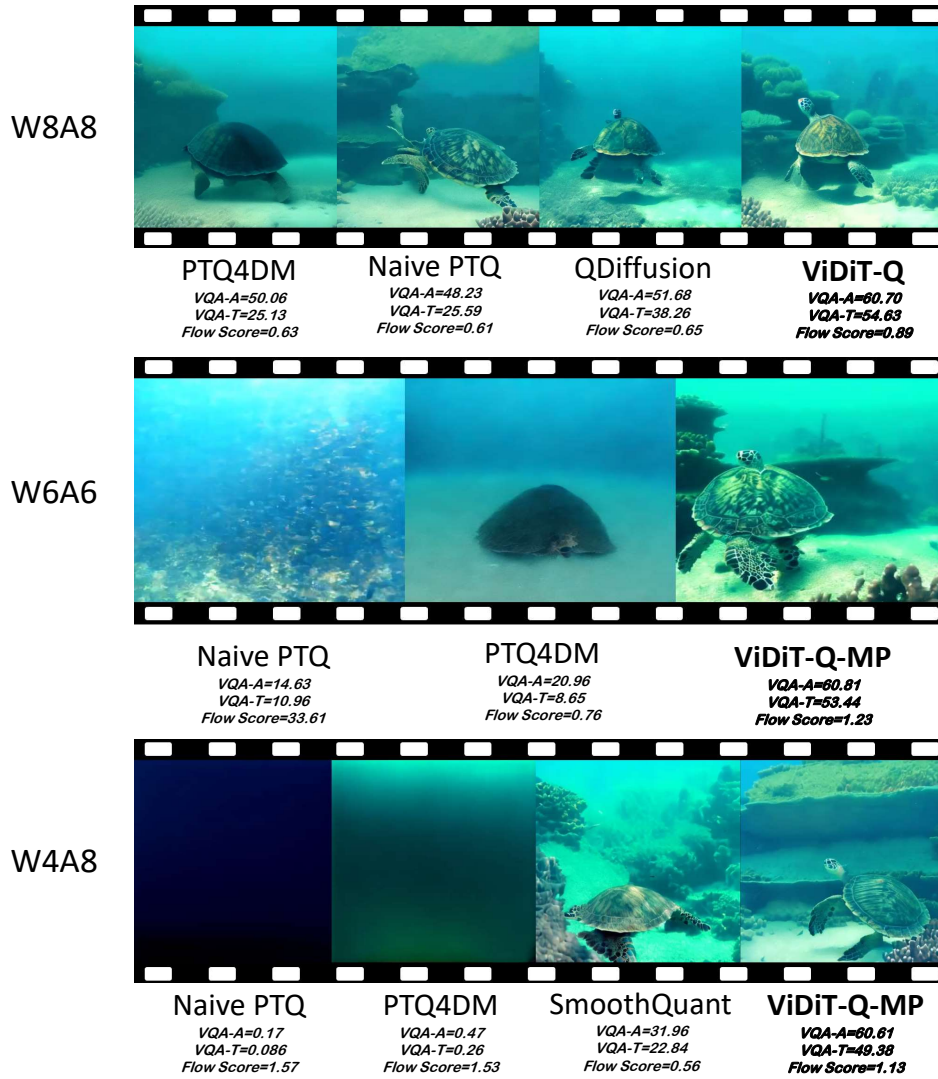


Figure 16: Example videos of different quantization methods for different bit-widths.

### C.3 Text-to-image Generation on COCO

We present more qualitative results of generated images by baseline quantization and ViDiT-Q quantization in Fig. 17. As shown, the Naïve PTQ’s generated images are highly blurred. While the W8A8 images depict outlines of objects, the W4A8 images generate nearly pure noises. In contrast, ViDiT-Q generates images nearly identical to the FP16 ones, preserving both visual quality and text-image alignment.

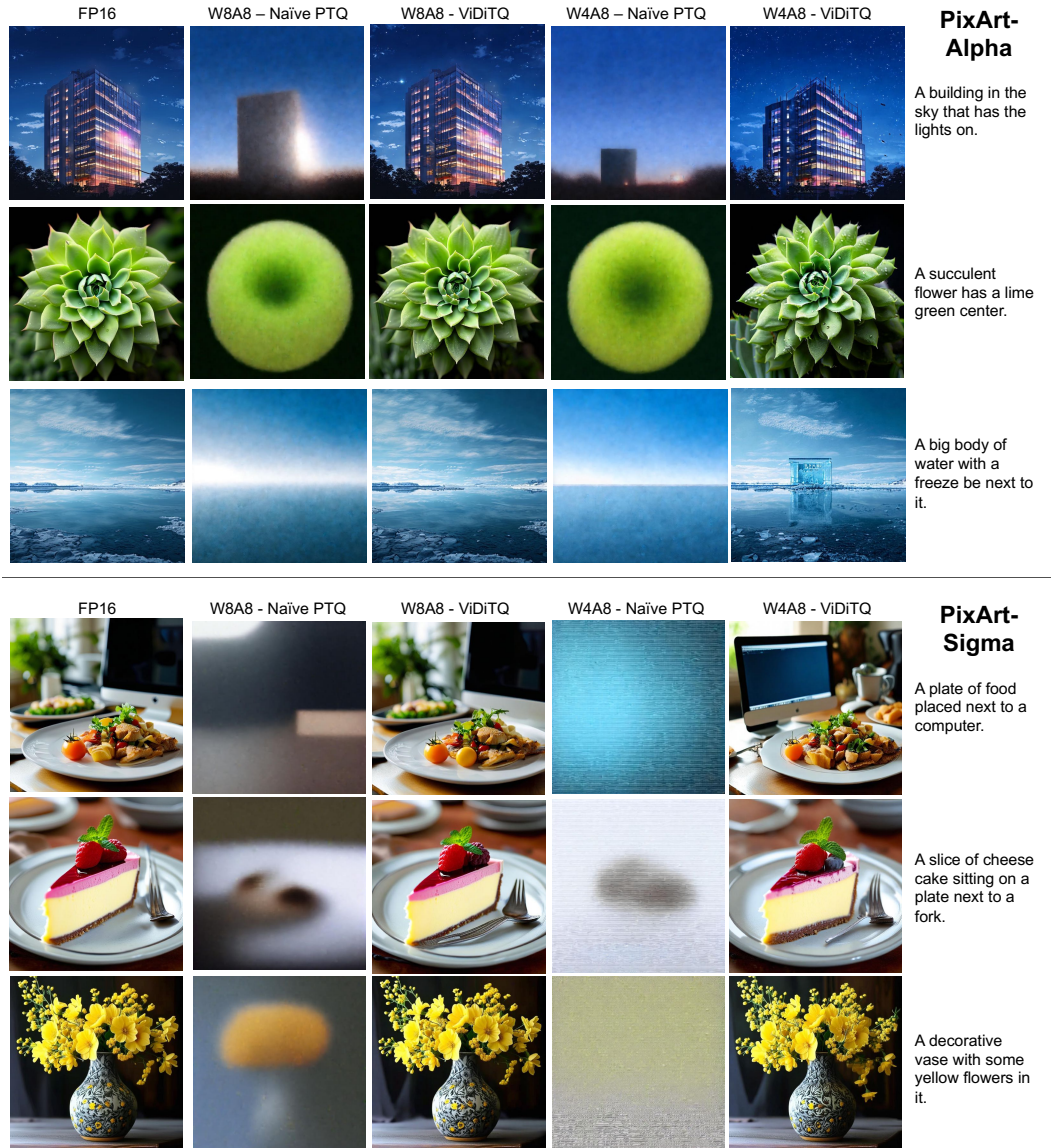


Figure 17: Qualitative results of text-to-image generation

## C.4 Ablation Studies

We present the generated videos for the ablation studies in Tab. 4. As seen in Fig. 18, video quality improves from blank images to similar to the FP16 baseline. For the challenging W4A8 quantization, the baseline method generates blank images. After adding dynamic quantization, some meaningful background (deep ocean) appears, but the main object (turtle) is still missing. Channel balancing reduces color deviation (from dark blue to green-blue), but the main object remains unrecognizable and changes significantly across frames (please refer to the supplementary materials for the video). Timestep-wise channel balancing improves the consistency of the main object, but notable degradation is still observed compared to the FP16 video. Finally, with mixed precision, a similar generation quality to the FP16 baseline is achieved."

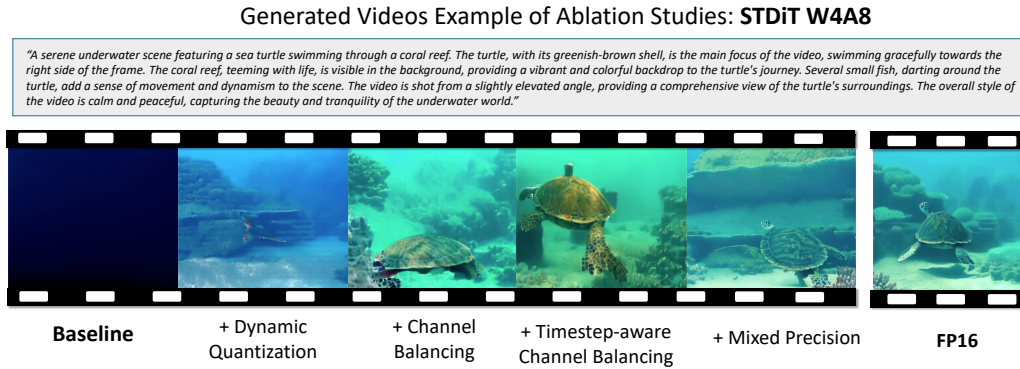


Figure 18: **Generated videos of ablation studies.**

We also present the comparison of quantization with naive channel balance and timestep-aware channel balance. As seen in Fig. 19, The main object in text instruction ("turtle" and "wispy grasses") in the left images are hardly recognizable. After introducing timestep-aware channel balancing, notable improvements are witnessed.

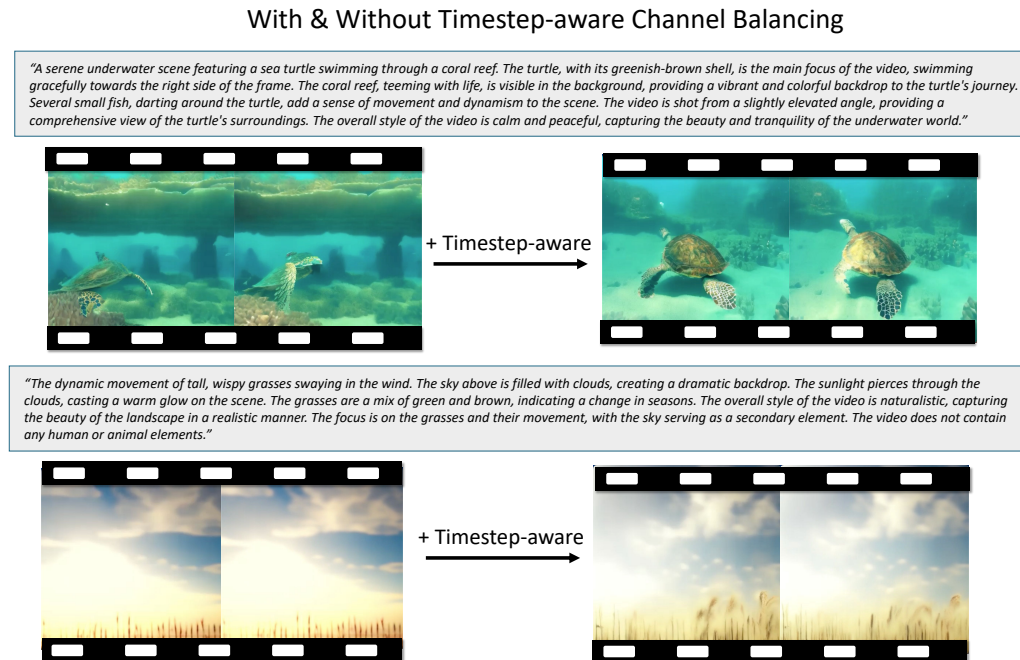


Figure 19: **Comparison of generated videos with naive and timestep-aware channel balancing.**