TR-DQ: Time-Rotation Diffusion Quantization

Yihua Shao[1,2,3], Deyang Lin[4], Minxi Yan[5], Siyu Chen[3], Fanhu Zeng[3], Minwen Liao[6], Ao Ma[7], Ziyang Yan[8], Haozhe Wang[9], Yan Wang[10], Zhi Chen[11], Xiaofeng Cao[12], Haotong Qin[13†], Hao Tang[1†], Jingcai Guo[2†]

[1]Peking University
[2]The Hong Kong Polytechnic University
[3]Institute of Automation, Chinese Academy of Sciences
[4]Guangdong University of Technology
[5]The Chinese University of Hong Kong
[6]Xinjiang University
[7]JD.com
[8]University of Trento
[9]The Hong Kong Univeristy of Science and Technology
[10]Tsinghua University
[11]University of Southern Queensland
[12]Tongji University
[13]ETH Zürich
yihuajerry@gmail.com, haotong.qin@pbl.ee.ethz.ch, haotang@pku.edu.cn, jc-jingcai.guo@polyu.edu.hk*

Abstract

Diffusion models have been widely adopted in image and video generation. However, their complex network architecture leads to high inference overhead for its generation process. Existing diffusion quantization methods primarily focus on the quantization of the model structure while ignoring the impact of time-steps variation during sampling. At the same time, most current approaches fail to account for significant activations that cannot be eliminated, resulting in substantial performance degradation after quantization. To address these issues, we propose **Time-Rotation Diffusion Quantization (TR-DQ)**, a novel quantization method incorporating time-step and rotation-based optimization. TR-DQ first divides the sampling process based on time-steps and applies a rotation matrix to smooth activations and weights dynamically. For different time-steps, a dedicated hyperparameter is introduced for adaptive timing modeling, which enables dynamic quantization across different time steps. Additionally, we also explore the compression potential of Classifier-Free Guidance (CFG-wise) to establish a foundation for subsequent work. TR-DQ achieves **state-of-the-art (SOTA)** performance on image generation and video generation tasks and a 1.38-1.89× speedup and 1.97-2.58× memory reduction in inference compared to existing quantization methods.

Introduction

Diffusion models (Ho et al. 2022; Xing et al. 2024) have demonstrated a remarkable ability to generate model parameters (Shao et al. 2025), 3D scenes (Erkoç et al. 2023; Wang et al. 2024; Yan et al. 2024b), etc. Also, they outperform GANs (Goodfellow et al. 2014, 2020) in most image and video generation tasks. However, due to their high memory consumption during inference, diffusion models are challenging to deploy on edge devices. In addition, the generation process consumes significant latency at each time-step, leading to low throughput, particularly for high-resolution images and long video generation. Therefore, compressing diffusion models while preserving their generative capability is crucial for practical deployment.

Several model compression methods are currently being tested in diffusion. In most of the model compression methods (Buciluǎ, Caruana, and Niculescu-Mizil 2006; Cheng et al. 2017; Zhu et al. 2024; Wang et al. 2025b). Quantization offers a promising solution to reduce memory and speed up computation for deployment on limited-resource devices. However, among them, post-training quantization (PTQ) methods (Frantar et al. 2022; Lin et al. 2024; Dettmers et al. 2023; Wang et al. 2025e) could avoid retraining the model, but do not achieve satisfactory results when applied directly to diffusion models. The distribution of time-steps significantly influences the generation process; hence, ignoring the activation distribution at each time-step can lead to negative effects. To address these issues, Q-Diffusion (Li et al. 2023) introduces a timestep-aware calibration and realizes end-to-end quantization, which enables the quantization of full-precision unconditional diffusion models into 4-bit. QUEST (Watson and Pelli 1983) identifies three key properties in quantized diffusion models affecting current methods: imbalanced activation distributions, imprecise temporal information, and specific module perturbation vulnerability. However, most of these methods mentioned above ignore the effect of significance activation in the diffusion model and therefore cause additional losses in quantification. In addition, Classifier-Free Guidance (CFG) is also a

*Correspondence to Haotong Qin, Hao Tang, and Jingcai Guo.

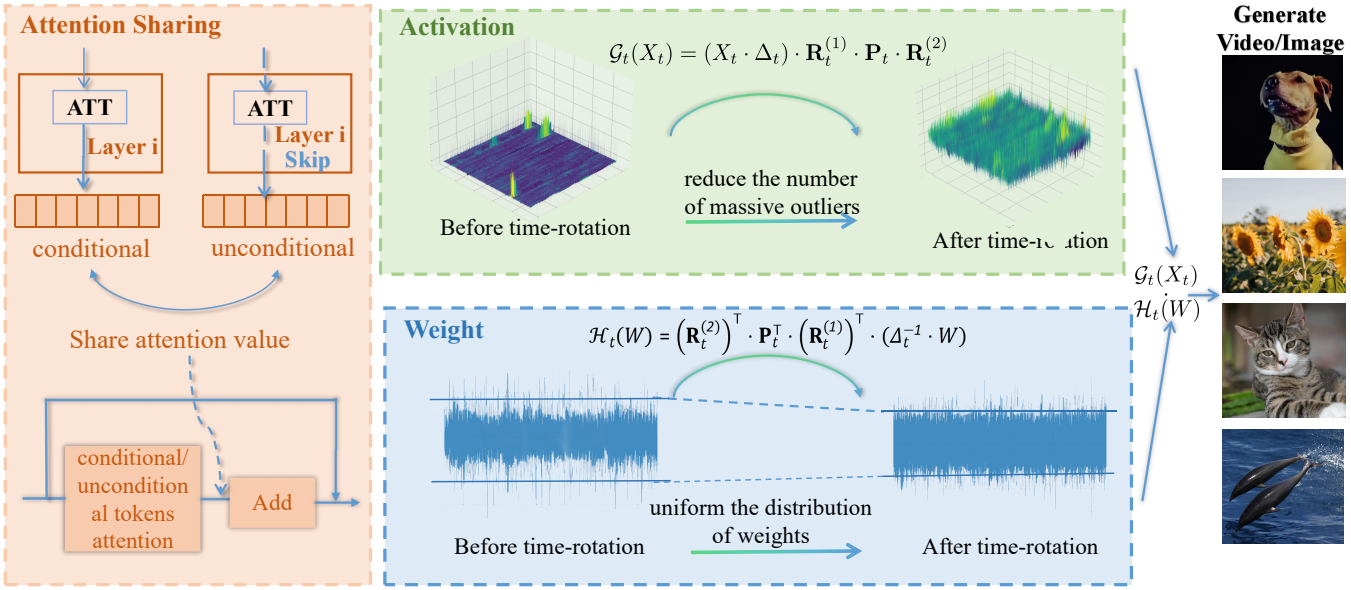


Figure 1: **Main pipeline of TR-DQ.** TR-DQ uses a rotation matrix for the activations to reduce the massive outliers, and also rearranges the weights to be a smoother and easier to quantify model overall. For CFG and non-CFG with high similarity of attention TR-DQ performs weight sharing, which further reduces the computational cost.

major factor that is ignored (Xie et al. 2024). To address these problems, we propose a time-step and rotation based quantization method, **Time-Rotation Diffusion Quantization (TR-DQ)**. TR-DQ first transfers the massive activations into weights using a rotation metrix, which makes activations and weights smoother and easier to quantize. Meanwhile, we explore both Classifier-Free Guidance (CFG) and non-CFG based quantization. Notably, we observe that some layers in CFG and non-CFG share similar parameter sensitivity distributions, allowing us to further compress them through a merging-based approach.

In order to demonstrate the effectiveness of our methodology, we conducted extensive experiments on image generation and video generation tasks. Experimental results demonstrate that our method outperforms existing quantization techniques in both image and video generation across most metrics, achieving state-of-the-art (SOTA) performance. Meanwhile, our method can achieve 1.7× speedup, significantly enhancing the efficiency of the generative model. The key contributions of our work are as follows:

- We shift the hard-to-quantify activations into weights using a rotation matrix, resulting in a smoother activation distribution that is easier to quantize.
- We introduce a novel quantization approach, Time-Rotation Diffusion Quantization (TR-DQ), by extending the global rotation matrix into a time-dependent rotation matrix based on the time-step distribution of diffusion models.
- By analyzing the similarity in attentional sensitivity between CFG and non-CFG, we implement attentional merging quantization to optimize compression.
- Our method significantly reduces quantization loss while preserving high visual quality in image and video generation tasks while achieving 1.38-1.89× speedup and 1.97-2.58× memory reduction without compromising performance.

Related Work

Generative Models

Image and video generation has achieved remarkable progress. Early GAN-based video generation methods (Gupta, Keshari, and Das 2022; Liu et al. 2021) have temporal coherence problems and consecutive frame discrepancies. Similarly, GAN-based image generation models (Karras, Laine, and Aila 2019) are known for their instability during training, frequently encountering problems such as mode collapseand. VAE-based methods (Yan et al. 2021; Kingma 2013; Yan et al. 2024a; Wang et al. 2025d) provide a robust framework but often require extensive computational resources. Video diffusion models with U-Net architecture were adapted to boost frame continuity. Latte (Ma et al. 2024) pioneered the use of transformer (Vaswani 2017) to realize high quality text-to-video generation, outperforming traditional methods in processing complex video data. SORA (Brooks et al. 2024; Zheng et al. 2024) further inspired the development of video diffusion transformer, advancing the development of models like GenTron (Chen et al. 2024b), which extended the capabilities of diffusion transformers to multi-frame video generation. Tora (Jankovic, Fontaine, and KokotoviC 1996) focuses on trajectory-oriented video generation and combines textual, visual, and trajectory conditions to create high-quality videos. However, existing image and video generation models still suffer from high memory cost. To address this issue, approaches such as model

quantization (Shao et al. 2024), pruning and distillation are proposed. In our work, we mainly focus on model quantization, and we will review milestones of diffusion quantization in Section Diffusion Model Quantization.

Diffusion Model Quantization

The evolution of model quantization has been instrumental in enabling the deployment of complex neural networks on resource-constrained devices. Post-training quantization (PTQ) methods like RTN (Nagel et al. 2020) and LLM.int8 (Dettmers et al. 2022) quantize weights and activations post-training with a few calibration dataset. However, most of the quantization methods are not suitable for diffusion models because diffusion models contain timesteps with different activation each steps. To address this, Q-Diffusion (Li et al. 2023) proposes a PTQ method for diffusion models, compressing them to 4-bit without performance loss by time-step-aware sampling and separation shortcut quantization. PTQ4DM (Shang et al. 2023) uses time-step-aware and separation shortcut techniques to compress models to 4-bit with similar performance to full-precision ones, and SVDQuant (Li et al. 2024) quantizes diffusion model weights and activations to 4-bit by introducing a low-rank branch to absorb outliers. Q-DiT (Chen et al. 2024a) customizes quantization parameters for channels to address imbalance, while PTQ4DiT (Wu et al. 2024) has designed a fixed mask adaptable to all timesteps to handle time-varying imbalance. For video generation model, ViDiT-Q (Zhao et al. 2024) designs a post-training quantization (PTQ) method for DiTs that enables W8A8 lossless quantization and W4A8 quantization without loss of generation quality. In our work, we mainly focus on time-steps modeling and explore the impact of Classifier-Free Guidance (CFG) wise.

Methodology

Preliminaries

As blocks of diffusion models are predominantly constructed with basic linear layers, which can be represented as, $Y = X \cdot W$. Here, W is the weight matrix. X and Y are denoted as input activations and output activations, respectively. In this paper, we focus on integer uniform quantization of both activations and weights, aiming to achieve better hardware support. Specifically, the b-bit quantization process maps the FP16 tensor X to low-bit integer X_{int} could be expressed as Eq. (1),

$$X_{\textbf{int}} = \text{clamp}\left(\left\lfloor \frac{X}{s} \right\rceil + z, 0, 2^b - 1\right), \tag{1}$$

where the function $clamp(x, 0, 2^b - 1)$ clamps the values x into range $[0, 2^b - 1]$, the nation $\lfloor \cdot \rceil$ means the nearest rounding operation. The scaling s could be expressed as Eq. (2),

$$s = \frac{\max(X) - \min(X)}{2^b - 1}. \tag{2}$$

and the zero point z could be calculated as Eq. (3),

$$z = -\left\lfloor \frac{\min(X)}{s} \right\rceil. \tag{3}$$

Following recent work, we employ per-token quantization for activations and per-channel quantization for weights.

For diffusion models, the presence of outliers in activations poses significant challenges to activations quantization. To address this issue, current quantization methods like SmoothQuant (Xiao et al. 2023) typically employ smoothing techniques, using computational invariance to shift the quantization difficulty from activations to weights. It's formula is as Eq. (8),

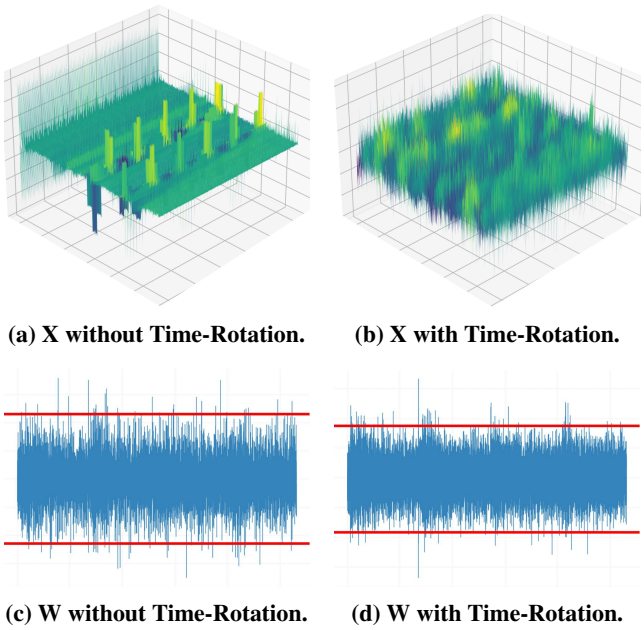
$$Y = \left(X \,\text{diag}(\Delta)^{-1}\right)\left(\,\text{diag}(\Delta)W\right) = \hat{X} \cdot \hat{W}. \tag{4}$$

The diagonal element Δ_j within Δ is computed as Eq. (5),

$$\Delta_j = \frac{\max\left(|X_j|\right)^{\alpha}}{\max\left(|W_j|\right)^{1-\alpha}}, \tag{5}$$

where α is a hyper-parameter representing the migration strength.

Quantization Strategies



(a) X without Time-Rotation. **(b) X with Time-Rotation.**

(c) W without Time-Rotation. **(d) W with Time-Rotation.**

Figure 2: **Effect of Time-Rotation on Data Distribution.** Data distribution with **Time-Rotation** is more smoother. Where X is the activations and W is the weights.

Examining Fig. 2a reveals that although smooth techniques reduce some outliers in activations, certain difficult to smooth outliers which we term **Massive Outliers** still persist. Although some outliers were smoothed, this did not change the unevenness of the data distribution (Wang et al. 2025c,a). All these factors affect quantization performance. Additionally, since the quantization difficulty is transferred from activations to weights, the weight distribution becomes even more irregular, making weight quantization another challenge (Shao et al. 2024). Therefore, adopting a novel

balancing strategy to equilibrate activations and weights is necessary. As it shown in Fig. 1, we leverage rotation matrices based on computational invariance. Through rotation matrices, we can reduce the number of **Massive Outliers** in activations and make the data distribution of both activations and weights more uniform, facilitating group-wise quantization. The specific details are as follows:

Balancing Strategies. Based on these observations and building upon DuQuant (Lin et al. 2025), we utilize an orthogonal rotation matrix R, a matrix constructed based on prior knowledge and greedy strategies, which can identify and swap the positions of outliers The construction of this rotation matrix is as Eq. (6),

$$\mathbf{R^1} = \mathbf{C_1 Q C_2}, \tag{6}$$

where the $\mathbf{C_1}$ is the switching matrix used to swap the first column and the column containing the maximum outlier columns of the activations, and $\mathbf{Q}$ represents an orthogonal randomly initialized rotation matrix, in which the first row is specifically uniformly distributed. The motivation behind this is to mitigate outliers in the first column after the transformation by $\mathbf{C_1}$. To ensure the orthogonality of the rotation matrix, we employ $\mathbf{C_2}$ to perform the corresponding row exchange that mirrors$\mathbf{C_1}$'s column operation. Specifically, if $\mathbf{C_1}$ swaps column 0 with column i (the outlier column), then $\mathbf{C_2}$ swaps row 0 with row i.

Thus, we obtain the final rotation matrix through a greedy strategy, with the formula as Eq. (7),

$$\mathbf{R} = \mathbf{R}^1\mathbf{R}^2 \cdots \mathbf{R}^n, \tag{7}$$

where $n = \arg\min_{k\in[1:N]} \left(max_{i,j}|(\mathbf{X}\mathbf{R}^1 \cdots \mathbf{R}^k)_{ij}|\right)$. Each $\mathbf{R}^i$ is constructed according to 6. Through this construction manner, we can ensure that the final rotation matrix $\mathbf{R}$ can effectively mitigate outliers with large magnitudes, as opposed to merely using a randomly selected orthogonal rotation matrix. Nevertheless, directly constructing the entire rotation matrix is time-consuming and results in substantial memory overhead. For fast matrix multiplication, following (Lin et al. 2025), we approximate the rotation matrix $\mathbf{R} \in \mathbb{R}^{C_{in}\times C_{in}}$ in a block-wise manner:

$$\mathbf{R} = \textbf{BlockDiag}(\mathbf{R}_{b_1}, ..., \mathbf{R}_{b_K}), \tag{8}$$

where $\mathbf{R}_{b_i} \in \mathbb{R}^{2^n\times 2^n}$ denotes a square matrix of the i-th block, which is constructed following the three steps mentioned above. And the block numbers K is calculated by $K = C_{in}/2^n$. After the first block rotation, most **Massive Outliers** can be eliminated, but in order to make the data smoother for per-token quantization, we need to perform a second block rotation. However, When we using the first block rotation reduces outliers locally, the distribution between different blocks may remain imbalanced, which is unfavorable for our second block rotation. To address this issue, we introduce the **zigzag permutation**. Concretely, we generate a zigzag sequence that starts by assigning channels with the highest activations to the first block. The process continues by assigning channels with the next highest activations to the subsequent blocks in descending order until the end of block K. Upon reaching the final block, the order reverses, starting from the channel with the next highest activations and proceeding in ascending order. This back-and-forth patterning continues throughout all the blocks, ensuring that no single block consistently receives either the highest or lowest activations channels. It is worth noting that the constructed permutation is also an orthogonal matrix, which we denote as $\mathbf{P}$. By employing the zigzag permutation, we achieve a balanced distribution of outliers across different blocks. This allows us to use the second block rotation to further smooth the outliers. The final balancing strategy can be represented as Eq. (9),

$$\begin{aligned} Y &= X \cdot W \\ &= \left[(X \cdot \Delta)\mathbf{R^{(1)}} \cdot \mathbf{P} \cdot \mathbf{R^{(2)}}\right] \cdot \\ &\quad \left[(\mathbf{R^{(2)}})^\top \cdot \mathbf{P}^\top \cdot (\mathbf{R^{(1)}})^\top (\Delta^{-1} \cdot W)\right], \end{aligned} \tag{9}$$

where the notation $\mathbf{P}$ denotes the orthogonal permutation matrix learned via the zigzag manner, the $\mathbf{R^{(1)}}$ and $\mathbf{R^{(2)}}$ represent the first and second block-diagonal rotation matrix, respectively. Through the application of the second rotation matrix, the activations values become smoother.

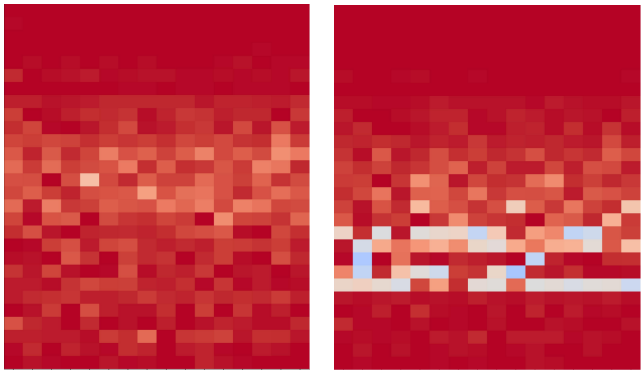
Time-Steps Awareness Quantization. Since activations at each time-step in diffusion models are different, applying a set of static quantization parameters to these activations would severely damage the generation quality of diffusion models. Furthermore, as activations at each time-step in diffusion models vary, the distribution of outliers in activations across different time-steps also differs significantly. If we still use a single set of $\mathbf{R}$, $\mathbf{P}$ and Δ for activations across all time-steps, this would ignore the distinctive characteristics of diffusion models and similarly impair their generation quality. To address these issues, we have implemented two approaches. First, we implement dynamic quantization for activations, calculating quantization parameters online. This process only requires additional computation of maximum and minimum values, making the computational cost negligible. Second, based on the characteristics of diffusion models, we propose **Time-Rotation**, which models the relationship between time and rotation matrices etc. Throughout the denoising process, instead of sharing a single set of $\mathbf{R}$, $\mathbf{P}$ and Δ, activations will select appropriate of $\mathbf{R}$, $\mathbf{P}$ and Δ based on the current time-step. Therefore, our final formula is as follow:

$$\begin{aligned} &F_t(X_t, W) = \mathcal{G}_t(X_t) \cdot \mathcal{H}_t(W), \\ &\mathcal{G}_t(X_t) = (X_t \cdot \Delta_t) \cdot \mathbf{R}_t^{(1)} \cdot \mathbf{P}_t \cdot \mathbf{R}_t^{(2)}, \\ &\mathcal{H}_t(W) = (\mathbf{R}_t^{(2)})^\top \cdot \mathbf{P}_t^\top \cdot (\mathbf{R}_t^{(1)})^\top \cdot (\Delta_t^{-1} \cdot W), \\ &F_t : \mathbb{R}^{T\times C_{in}} \times \mathbb{R}^{C_{in}\times C_{out}} \to \mathbb{R}^{T\times C_{out}}, \\ &t \in T = \{1, 2, \ldots, N\}. \end{aligned} \tag{10}$$

Compared to ViDiTQ (Zhao et al. 2024)'s Quarot (Ashkboos et al. 2024) based rotation matrices, our **Time-Rotation** are more sophisticated in construction. Rather than being simply initialized, they are constructed based on prior knowledge and greedy strategies. In addition, the extra permutation operation enables better handling of unevenly distributed data. As demonstrated in 2b and 2d, this

method effectively smooths the distribution of both activations and weights, with the activations values range narrowing from [-2.0, 1.5] to [-0.6, 0.6], and the weight values become smoother. This evidence confirms that our method achieves superior smoothing effects compared to conventional smoothing approaches.

Attention-Sharing Quantization (AS)



(a) Heat map at time-step=12. (b) Heat map at time-step=15.

Figure 3: **Heat maps of multi-head self-attention under conditional and unconditional situations.** Each square reflects the similarity between the two. The redder the square, the higher the similarity; the bluer the square, the lower the similarity.

Classifier-free guidance (CFG) is widely used for diffusion transformers, enabling the generation of more imaginative images or videos that are not confined to a single format. However, the adoption of CFG technology means that we cannot complete the task with just a single denoising process, which significantly slows down inference speed. To address this issue, through examination of the structure of diffusion transformers, we discovered that there exists substantial similarity between attention values of multi-head self-attention in each block of the condition and unconditional paths, as shown in 3a. Our evaluation metric is cosine similarity, detailed as follows Eq. (11),

$$\cos(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \, \|\mathbf{B}\|}. \tag{11}$$

However, due to the time-step differences mentioned earlier, not all attention values are identical across every time-step, as illustrated in 3b. Through extensive observation, we found that the attention values of multi-head self-attention in the forward and backward blocks show significant similarities across all time-steps. Therefore, we decided to implement Attention-Sharing for these blocks. Subsequent experiments demonstrated that our Attention-Sharing approach improves inference speed without compromising the quality of generated images or videos.

Experiments

Experiment Settings

Setting up. We follow the experimental configuration of VIDITQ (Zhao et al. 2024), employing PixArt-α (Chen et al. 2023) (pre-trained on COCO (Lin et al. 2014)) and Open-SORA (Zheng et al. 2024) (pre-trained on UCF-101 (Soomro, Zamir, and Shah 2012)) for image and video generation tasks, respectively. The parameters are set to STEP=20 and CFG=4.5. All experiments are conducted on an NVIDIA A800 GPU (80GB).

Baselines. Since many diffusion quantization methods are developed from LLMs quantization methods, our baseline includes diffusion quantization methods and LLMs quantization methods. Therefore, the quantization schemes for LLMs that we have chosen include SmoothQuant (Xiao et al. 2023), DuQuant (Lin et al. 2025), and Quarot (Ashkboos et al. 2024), all of which can achieve weight-activation quantization. The selected diffusion quantization baselines include Q-Diffusion (Li et al. 2023), Q-DiT (Chen et al. 2024a), PTQ4DiT (Wu et al. 2025) and ViDiT-Q (Zhao et al. 2024).

Evaluation Metrics. For image generation, we adopt FID (Heusel et al. 2017), CLIPScore (Hessel et al. 2021), and ImageReward (Xu et al. 2023). Video generation is evaluated using VBench (Huang et al. 2024). To assess the alignment between language and video, we follow the metrics introduced in EvalCrafter (Liu et al. 2024). Model efficiency is measured in terms of inference peak memory and throughput. Detailed definitions of all evaluation metrics are provided in the supplementary material.

Main Results

Image Generation Tasks. In this section, we will discuss the superiority of our method through the existing benchmark and visualization results. To ensure comprehensiveness, we will introduce some of the existing quantization methods for large language models (LLMs) as baseline for comparison. As it shown in Tab. 2, the model quantinaized by TR-DQ leads the model quantinaized by the other quantitative methods in all metrics. At the same number of quantization bits, both TR-DQ and the method after weight sharing generate images with better quality than ViDiT-Q. This suggests that many of the previous methods ignored the time-steps change of MASSIVE activation and activation, while TR-DQ proposed a more effective solution. When the weights were quantinaized to 4 bits, the effect of massive activation was ignored as ViDiT only smoothed the outliers of the weights by rotating the matrix. Therefore, in the case of diffusion quantization, the treatment of activations greatly affects the performance of the quantinaized model. In contrast, our approach focuses on the activations, which are smoother and more favourable for quantization.

Video Generation Tasks. The results of the experimental video generation evaluation may have some errors due to the poor robustness of the benchmark vbench. As it shown in Tab. 1, TR-DQ is better than the current SOTA ViDiTQ in most metrics. especially when the weights are quantized to 4bit. This difference is more significant compared to im-

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	97.01	56.94	96.13	92.28	40.51	26.21
Q-Diffusion (Li et al. 2023)	8/8	60.38	55.15	94.44	68.05	94.17	87.74	36.62	25.66
Q-DiT (Chen et al. 2024a)	8/8	60.35	55.80	93.64	68.05	94.70	86.94	32.34	26.09
PTQ4DiT (Wu et al. 2024)	8/8	56.88	55.53	95.89	63.88	96.02	91.26	34.52	25.32
ViDiT-Q (Zhao et al. 2024)	8/8	61.48	56.95	96.14	61.11	95.84	90.24	38.22	26.06
TR-DQ (Ours)	8/8	61.82	57.44	96.63	55.38	96.11	91.14	39.78	26.18
TR-DQ+AS (Ours)	8/8	60.38	57.10	96.26	50.27	95.71	91.58	38.50	25.99
Q-DiT (Chen et al. 2024a)	4/8	23.30	29.61	97.89	4.166	97.02	91.51	0.00	4.985
PTQ4DiT (Wu et al. 2024)	4/8	37.97	31.15	92.56	9.722	98.18	93.59	3.561	11.46
ViDiT-Q(Zhao et al. 2024)	4/8	59.01	55.37	95.69	48.33	95.23	88.72	36.19	25.94
TR-DQ (Ours)	4/8	59.88	56.20	96.57	51.83	96.65	90.74	32.46	26.17
TR-DQ+AS (Ours)	4/8	57.69	55.02	96.74	47.78	96.58	91.03	32.25	25.34

Table 1: **Performance of TR-DQ on video generation on VBench.** TR-DQ outperforms the current SOTA ViDiTQ in most metrics, suggesting that it is more capable of generating models after quantization.

Method	Bit-width (W/A)	FID(↓)	CLIP(↑)	IR(↑)
-	16/16	73.34	0.258	0.901
Q-Diffusion (Li et al. 2023)	8/8	96.54	0.239	0.186
Q-DiT (Chen et al. 2024a)	8/8	73.60	0.256	0.854
PTQ4DiT (Wu et al. 2024)	8/8	127.9	0.217	-1.216
ViDiT-Q (Zhao et al. 2024)	8/8	75.98	0.232	0.859
TR-DQ (Ours)	8/8	75.12	0.249	0.887
TR-DQ+AS (Ours)	8/8	75.57	0.233	0.863
Q-Diffusion (Li et al. 2023)	4/8	91.95	0.228	-0.224
Q-DiT (Chen et al. 2024a)	4/8	475.8	0.127	-2.277
PTQ4DiT (Wu et al. 2024)	4/8	171.9	0.177	-2.064
ViDiT-Q (Zhao et al. 2024)	4/8	76.65	0.243	0.837
TR-DQ (Ours)	4/8	75.53	0.252	0.851
TR-DQ+AS (Ours)	4/8	75.76	0.246	0.847

Table 2: **Results of image generation task.** TR-DQ method has an overall advantage over current quantization methods for the same bits. AS indicates Attention-Sharing. In addition, it is worth noting that **ViDiT-Q's W4A8 uses a mixed quantization** that means the weights are not really 4-bit quantization, there may be 6 and 8 bits.

ages, and we believe that one reason is that the video generation model has a larger number of parameters compared to the image generation model, so the effect of activation quantization is more significant. In addition, the video generation task has a stronger timing dependency than the image generation task, so other quantization methods may lack fine-grained timing divisions, leading to poorer quality of the generated video. As shown in Tab. 3, for most metrics, TR-DQ is ahead of other methods. This suggests that the video-text consistency of the quantized TR-DQ model has an advantage over other methods. And this gap becomes more obvious when the weights are quantized to 4bit. This shows that we effectively shift the activation to the weights to reduce the error caused by the activation. Also, we propose a long prompt video generation sample to for visualisation. The video quality generated by the compressed model is slightly degraded compared to the original model, but still maintains better results. The comparison with the quantization approach to large language model can be found in the supplementary material.

Classifier-Free Guidance Result. As it shown in Tab. 2, with the addition of CFG weight sharing in TR-DQ, although there is a loss in the quality of the generated image, its effect still outperforms the current SOTA method. Therefore, it indicates that the difference between partial CFG and non-CFG attention distribution is not obvious, and there is a possibility of compression. As it shown in Tab. 1 and Tab. 3, Attention Sharing leads to a decrease in the quality of video generation, but the generative power of the model still differs little from the current SOTA. Therefore, our approach reduces redundant ATTENTION computations while maintaining model generation capabilities.

Efficiency Comparison. As it shown in Tab. 5, TR-DQ significantly reduces the memory overhead and latency compared to the original model, and is more conducive to hardware inference computation because the TR-DQ activation and weight distributions are smoother than those of ViDiTQ. Further, we reduce the attention computation by weight sharing so that the model can skip the layers with high CFG, no-CFG similarity in inference. This operation reduces the overall inference overhead of the model and also reduces the computational latency of the model.

Ablation Study

In this section, we discuss the main influences of our methodology. We will discuss our main contribution from the rotation matrix approach and time modeling.

As it shown in Tab. 4, $\mathbf{R_1}$ is the rotation matrix for handling activation outliers, $\mathbf{R_2}$ is the rotation matrix for handling weight outliers, and $\mathbf{P}$ denotes the weight permutation process. We found that the biggest impact on the model was the dynamic transformation of the rotation matrix based on time-steps. The overall quality of the generated video is significantly improved by adding time information. The fact

Method	Bit-width (W/A)	CLIPSIM	CLIP-Temp	VQA-Aesthetic	VQA-Technical	Δ Flow Score. (↓)
-	16/16	0.1818	0.9988	63.40	50.46	-
Q-Diffusion (Li et al. 2023)	8/8	0.1781	0.9987	51.68	38.27	0.328
Q-DiT (Chen et al. 2024a)	8/8	0.1788	0.9977	61.03	34.97	0.473
PTQ4DiT (Wu et al. 2024)	8/8	0.1836	0.9991	54.56	53.33	0.440
ViDiT-Q (Zhao et al. 2024)	8/8	0.1950	0.9991	60.70	54.64	0.089
TR-DQ (Ours)	8/8	0.1861	0.9990	62.43	57.07	0.295
TR-DQ+AS (Ours)	8/8	0.1830	0.9991	59.16	51.52	0.128
Q-DiT (Chen et al. 2024a)	6/6	0.1710	0.9943	11.04	1.869	41.10
PTQ4DiT (Wu et al. 2024)	6/6	0.1799	0.9976	59.97	43.89	0.997
ViDiT-Q (Zhao et al. 2024)	6/6	0.1791	0.9984	64.45	51.58	0.625
TR-DQ (Ours)	6/6	0.1795	0.9988	61.80	49.58	0.042
TR-DQ+AS (Ours)	6/6	0.1747	0.9987	59.74	44.94	0.068
Q-DiT (Chen et al. 2024a)	4/8	0.1687	0.9833	0.007	0.018	3.013
PTQ4DiT (Wu et al. 2024)	4/8	0.1735	0.9973	2.210	0.318	0.108
ViDiT-Q (Zhao et al. 2024)	4/8	0.1809	0.9989	60.62	49.38	0.153
TR-DQ (Ours)	4/8	0.1815	0.9990	59.86	55.56	0.130
TR-DQ+AS (Ours)	4/8	0.1715	0.9993	56.87	48.09	0.306

Table 3: **The corresponding effects of different quantization methods on prompt.** Video generated at different bit-widths with response to prompt. q-diffusion does not generate video properly at W6A6 and W4A8.

Methods					Bit-width	CLIPSIM	CLIP-Temp	VQA-	VQA-	Δ Flow
Smooth	$\mathbf{R_1}$	P	$\mathbf{R_2}$	T-R	(W/A)			Aesthetic	Technical	Score.
-	-	-	-	-	16/16	0.1797	0.9988	63.40	50.46	-
✓	-	-	-	-	4/8	0.1739	0.9985	44.12	21.19	0.675
✓	✓	-	-	-	4/8	0.1755	0.9941	50.42	42.12	0.421
✓	✓	-	✓	-	4/8	0.1745	0.9972	52.38	45.67	0.342
✓	✓	✓	✓	-	4/8	0.1741	0.9985	54.94	47.97	0.219
✓	✓	✓	✓	✓	4/8	0.1815	0.9990	59.86	55.56	0.130

Table 4: **Ablation studies of TR-DQ.** We discuss the main influences on the model when quantifying W4A8.

Bit-width (W/A)	A800		A100	
	Memory	Latency	Memory	Latency
16/16	1.00×	1.00×	1.00×	1.00×
8/8 (ViDiTQ)	1.98×	1.70×	2.00×	1.74×
8/8 (TR-DQ)	1.97×	1.69×	1.97×	1.69×
8/8 (TR-DQ+AS)	2.17×	1.89×	2.17×	1.91×
4/8 (ViDiTQ)	2.41×	1.36×	2.42×	1.38×
4/8 (TR-DQ)	2.46×	1.38×	2.47×	1.42×
4/8 (TR-DQ+AS)	2.58×	1.41×	2.59×	1.44×

Table 5: **Efficiency Comparison between original model and SOTA method.** The size and lantency of the compressed model of TR-DQ is almost the same as that of ViDiTQ.

that $\mathbf{R_1}$ has a greater impact on the overall effect of the video than any other factor suggests that diffusion is different from the large language model in that it directly affects the quality of the generation of the generative model. Permutation and $\mathbf{R_2}$, although both affect the video generation results, are not major factors. In contrast $\mathbf{R_2}$ has a greater impact than permutation.

Conclusion

In this article, we explore two current issues in diffusion model quantization: massive activation and time-steps sampling. To address these problems, we design a rotation matrix quantization method based on time-steps activation distribution, **Time-Rotation Diffusion Quantization (TR-DQ)**. TR-DQ shifts hard to quant activations to weights via a matrix and adaptively adjusts the parameters of the rotation matrix for each time-step activation change. Meanwhile, we found that some layers have higher weight similarity in the case of CFG and non-CFG, so we chose to merge these weights for processing to reduce the memory overhead of CFG. Our method has better image and video generation compared to current quantization methods. Compared to original model, our approach achieve 1.38-1.89× speedup and 1.97-2.58× memory reduction.

Acknowledgments

This work was supported in part by the Fundamental Research Funds for the Central Universities (Peking Univer-

sity), the Hong Kong RGC General Research Fund (Nos. 15221123 and 15216424), the PolyU Internal Fund (No. P0058468), and the Huawei Gifted Fund. We also acknowledge the support from the Swiss National Science Foundation (SNSF) through project 200021E_219943 (Neuromorphic Attention Models for Event Data).

References

Ashkboos, S.; Mohtashami, A.; Croci, M. L.; Li, B.; Cameron, P.; Jaggi, M.; Alistarh, D.; Hoefler, T.; and Hensman, J. 2024. Quarot: Outlier-free 4-bit inference in rotated llms. *arXiv preprint arXiv:2404.00456*.

Brooks, T.; Peebles, B.; Holmes, C.; DePue, W.; Guo, Y.; Jing, L.; Schnurr, D.; Taylor, J.; Luhman, T.; Luhman, E.; et al. 2024. Video generation models as world simulators.

Buciluǎ, C.; Caruana, R.; and Niculescu-Mizil, A. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 535–541.

Chen, J.; Yu, J.; Ge, C.; Yao, L.; Xie, E.; Wu, Y.; Wang, Z.; Kwok, J.; Luo, P.; Lu, H.; et al. 2023. Pixart-$alpha$: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*.

Chen, L.; Meng, Y.; Tang, C.; Ma, X.; Jiang, J.; Wang, X.; Wang, Z.; and Zhu, W. 2024a. Q-dit: Accurate post-training quantization for diffusion transformers. *arXiv preprint arXiv:2406.17343*.

Chen, S.; Xu, M.; Ren, J.; Cong, Y.; He, S.; Xie, Y.; Sinha, A.; Luo, P.; Xiang, T.; and Perez-Rua, J.-M. 2024b. GenTron: Diffusion Transformers for Image and Video Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6441–6451.

Cheng, Y.; Wang, D.; Zhou, P.; and Zhang, T. 2017. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282*.

Dettmers, T.; Lewis, M.; Belkada, Y.; and Zettlemoyer, L. 2022. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35: 30318–30332.

Dettmers, T.; Svirschevski, R.; Egiazarian, V.; Kuznedelev, D.; Frantar, E.; Ashkboos, S.; Borzunov, A.; Hoefler, T.; and Alistarh, D. 2023. Spqr: A sparse-quantized representation for near-lossless llm weight compression. *arXiv preprint arXiv:2306.03078*.

Erkoç, Z.; Ma, F.; Shan, Q.; Nießner, M.; and Dai, A. 2023. Hyperdiffusion: Generating implicit neural fields with weight-space diffusion. In *Proceedings of the IEEE/CVF international conference on computer vision*, 14300–14310.

Frantar, E.; Ashkboos, S.; Hoefler, T.; and Alistarh, D. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

Gupta, S.; Keshari, A.; and Das, S. 2022. Rv-gan: Recurrent gan for unconditional video generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024–2033.

Hessel, J.; Holtzman, A.; Forbes, M.; Bras, R. L.; and Choi, Y. 2021. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30.

Ho, J.; Salimans, T.; Gritsenko, A.; Chan, W.; Norouzi, M.; and Fleet, D. J. 2022. Video diffusion models. *Advances in Neural Information Processing Systems*, 35: 8633–8646.

Huang, Z.; He, Y.; Yu, J.; Zhang, F.; Si, C.; Jiang, Y.; Zhang, Y.; Wu, T.; Jin, Q.; Chanpaisit, N.; et al. 2024. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21807–21818.

Jankovic, M.; Fontaine, D.; and KokotoviC, P. V. 1996. TORA example: cascade-and passivity-based control designs. *IEEE Transactions on Control Systems Technology*, 4(3): 292–297.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4401–4410.

Kingma, D. P. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Li, M.; Lin, Y.; Zhang, Z.; Cai, T.; Li, X.; Guo, J.; Xie, E.; Meng, C.; Zhu, J.-Y.; and Han, S. 2024. Svdqunat: Absorbing outliers by low-rank components for 4-bit diffusion models. *arXiv preprint arXiv:2411.05007*.

Li, X.; Liu, Y.; Lian, L.; Yang, H.; Dong, Z.; Kang, D.; Zhang, S.; and Keutzer, K. 2023. Q-diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 17535–17545.

Lin, H.; Xu, H.; Wu, Y.; Cui, J.; Zhang, Y.; Mou, L.; Song, L.; Sun, Z.; and Wei, Y. 2025. Duquant: Distributing outliers via dual transformation makes stronger quantized llms. *Advances in Neural Information Processing Systems*, 37: 87766–87800.

Lin, J.; Tang, J.; Tang, H.; Yang, S.; Chen, W.-M.; Wang, W.-C.; Xiao, G.; Dang, X.; Gan, C.; and Han, S. 2024. AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration. *Proceedings of Machine Learning and Systems*, 6: 87–100.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*, 740–755. Springer.

Liu, M.-Y.; Huang, X.; Yu, J.; Wang, T.-C.; and Mallya, A. 2021. Generative adversarial networks for image and video synthesis: Algorithms and applications. *Proceedings of the IEEE*, 109(5): 839–862.

Liu, Y.; Cun, X.; Liu, X.; Wang, X.; Zhang, Y.; Chen, H.; Liu, Y.; Zeng, T.; Chan, R.; and Shan, Y. 2024. Evalcrafter: Benchmarking and evaluating large video generation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22139–22149.

Ma, X.; Wang, Y.; Jia, G.; Chen, X.; Liu, Z.; Li, Y.-F.; Chen, C.; and Qiao, Y. 2024. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*.

Nagel, M.; Amjad, R. A.; Van Baalen, M.; Louizos, C.; and Blankevoort, T. 2020. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, 7197–7206. PMLR.

Shang, Y.; Yuan, Z.; Xie, B.; Wu, B.; and Yan, Y. 2023. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1972–1981.

Shao, Y.; Liang, S.; Ling, Z.; Yan, M.; Liu, H.; Chen, S.; Yan, Z.; Zhang, C.; Qin, H.; Magno, M.; et al. 2024. GWQ: Gradient-Aware Weight Quantization for Large Language Models. *arXiv preprint arXiv:2411.00850*.

Shao, Y.; Yan, M.; Liu, Y.; Chen, S.; Chen, W.; Long, X.; Yan, Z.; Li, L.; Zhang, C.; Sebe, N.; et al. 2025. In-Context Meta LoRA Generation. *arXiv preprint arXiv:2501.17635*.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Vaswani, A. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Wang, H.; Li, L.; Qu, C.; Zhu, F.; Xu, W.; Chu, W.; and Lin, F. 2025a. To code or not to code? adaptive tool integration for math language models via expectation-maximization. *arXiv preprint arXiv:2502.00691*.

Wang, H.; Qu, C.; Huang, Z.; Chu, W.; Lin, F.; and Chen, W. 2025b. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *arXiv preprint arXiv:2504.08837*.

Wang, H.; Que, H.; Xu, Q.; Liu, M.; Zhou, W.; Feng, J.; Zhong, W.; Ye, W.; Yang, T.; Huang, W.; et al. 2025c. Reverse-Engineered Reasoning for Open-Ended Generation. *arXiv preprint arXiv:2509.06160*.

Wang, H.; Su, A.; ; Ren, W.; Lin, F.; and Chen, W. 2025d. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *arXiv preprint arXiv:2505.15966*.

Wang, H.; Xu, Q.; Liu, C.; Wu, J.; Lin, F.; and Chen, W. 2025e. Emergent Hierarchical Reasoning in LLMs through Reinforcement Learning. *arXiv preprint arXiv:2509.03646*.

Wang, L.; Zheng, W.; Ren, Y.; Jiang, H.; Cui, Z.; Yu, H.; and Lu, J. 2024. Occsora: 4d occupancy generation models as world simulators for autonomous driving. *arXiv preprint arXiv:2405.20337*.

Watson, A. B.; and Pelli, D. G. 1983. QUEST: A Bayesian adaptive psychometric method. *Perception & psychophysics*, 33(2): 113–120.

Wu, J.; Wang, H.; Shang, Y.; Shah, M.; and Yan, Y. 2024. PTQ4DiT: Post-training Quantization for Diffusion Transformers. *arXiv preprint arXiv:2405.16005*.

Wu, J.; Wang, H.; Shang, Y.; Shah, M.; and Yan, Y. 2025. Ptq4dit: Post-training quantization for diffusion transformers. *Advances in Neural Information Processing Systems*, 37: 62732–62755.

Xiao, G.; Lin, J.; Seznec, M.; Wu, H.; Demouth, J.; and Han, S. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, 38087–38099. PMLR.

Xie, R.; Zhao, T.; Yuan, Z.; Wan, R.; Gao, W.; Zhu, Z.; Ning, X.; and Wang, Y. 2024. LiteVAR: Compressing Visual Autoregressive Modelling with Efficient Attention and Quantization. *arXiv preprint arXiv:2411.17178*.

Xing, Z.; Feng, Q.; Chen, H.; Dai, Q.; Hu, H.; Xu, H.; Wu, Z.; and Jiang, Y.-G. 2024. A survey on video diffusion models. *ACM Computing Surveys*, 57(2): 1–42.

Xu, J.; Liu, X.; Wu, Y.; Tong, Y.; Li, Q.; Ding, M.; Tang, J.; and Dong, Y. 2023. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36: 15903–15935.

Yan, W.; Zhang, Y.; Abbeel, P.; and Srinivas, A. 2021. Videogpt: Video generation using vq-vae and transformers. *arXiv preprint arXiv:2104.10157*.

Yan, Z.; Dong, W.; Shao, Y.; Lu, Y.; Haiyang, L.; Liu, J.; Wang, H.; Wang, Z.; Wang, Y.; Remondino, F.; et al. 2024a. Renderworld: World model with self-supervised 3d label. *arXiv preprint arXiv:2409.11356*.

Yan, Z.; Li, L.; Shao, Y.; Chen, S.; Kai, W.; Hwang, J.-N.; Zhao, H.; and Remondino, F. 2024b. 3dsceneeditor: Controllable 3d scene editing with gaussian splatting. *arXiv preprint arXiv:2412.01583*.

Zhao, T.; Fang, T.; Liu, E.; Wan, R.; Soedarmadji, W.; Li, S.; Lin, Z.; Dai, G.; Yan, S.; Yang, H.; et al. 2024. Vidit-q: Efficient and accurate quantization of diffusion transformers for image and video generation. *arXiv preprint arXiv:2406.02540*.

Zheng, Z.; Peng, X.; Yang, T.; Shen, C.; Li, S.; Liu, H.; Zhou, Y.; Li, T.; and You, Y. 2024. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*.

Zhu, X.; Li, J.; Liu, Y.; Ma, C.; and Wang, W. 2024. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12: 1556–1577.