QUANT*WM*: TEMPORALLY CONSISTENT 2-BIT KV CACHE QUANTIZATION FOR VIDEO WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Video generation based world models achieve long-range temporal consistency by storing KV cache during generation, but the continuously growing cache makes KV cache memory become a major deployment bottleneck, which motivates low-bit quantization study for efficiency. Existing 2-bit KV cache quantization methods can achieve nearly quantitatively lossless performance on conventional video benchmarks such as VBench, however, when applied to video world models, we find that they still cause severe temporal flickering and visual degradation. Meanwhile, deeper investigates show that Key quantization produces smaller reconstruction errors than Value, but surprisingly leads to much larger output degradation. We trace this discrepancy to attention in video world models: Key perturbations can change the attention logits, *i.e.*, $QK^\top$, and shift the temporal-spatial tokens selected by Queries. These observations motivate us to explicitly preserve attention logits and temporal-spatial token selection during KV cache quantization to alleviate the visual degradation problem. To address this issue, we present **Quant*WM***, a training-free 2-bit KV cache quantization framework for video world models. **Quant*WM*** introduces two complementary techniques to mitigate the attention shifts. Firstly, *quantization-sensitivity-aware clustering (QSAC)* jointly considers historical Query sensitivity and residual ranges to select INT2-friendly Key centroids, which reduces quantization errors in channels that are more critical to attention. In addition, *principal-subspace attention compensation (PSAC)* restores the remaining Key errors along the dominant Query subspace using low-rank projections, which provides a direct and efficient correction to stabilize attention logits. Extensive experiments on LingBot-World-v2, HY-World 1.5, Matrix-Game-2, Longcat-Video and Causal-Forcing demonstrate that **Quant*WM*** significantly improves visual quality and temporal consistency, while outperforming existing methods across image and video quality metrics with up to $6.20\times$ KV cache memory compression and limited additional overhead.

1 INTRODUCTION

Recent advances in video generation (Weissenborn et al., 2019; Singer et al., 2022; Zheng et al., 2024; Yang et al., 2025b) have achieved increasingly realistic and temporally consistent visual synthesis, while video world models (Alonso et al., 2024; Xiao et al., 2026; Wang et al., 2026) further extend these capabilities to interactive world simulation conditioned on user actions and camera trajectories. In causal and autoregressive video world models (Huang et al., 2026; Liu et al., 2026), KV cache stores historical Key and Value representations, which allows each generated chunk to attend to relevant world context from previous chunks, so that achieving long-range temporal consistency while supporting continuous interaction. However, the large number of spatial-temporal tokens makes the KV cache extremely memory-intensive and becomes a major bottleneck for practical deployment. For example, LingBot-World-v2 (Gao et al., 2026) requires more than 21 GiB of KV cache memory to generate a 5-second video (about 93 frames), which significantly limits deployment on memory-constrained devices.

Quantization (Frantar et al., 2022; Lin et al., 2024; Xiao et al., 2023; Zhao et al., 2024) is an effective model compression technique which represents high-precision tensors with low-bit values. For KV caches, quantization directly reduces the storage cost of cached Keys and Values, and has achieved

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



Figure 1: Existing 2-bit KV cache quantization methods achieve nearly lossless video performance but do not fully capture temporal flickering and visual degradation. **Quant*WM*** effectively improves temporal consistency and significantly reduce the KV cache memory for video world models.

promising results in large language models (Liu et al., 2024b; Hooper et al., 2024; Lin et al., 2025). Particularly, recent methods can compress KV caches to 2-bit or lower with limited performance degradation (Zhang et al., 2024; Li et al., 2025). Building on these advances, Quant-VideoGen (QVG) (Xi et al., 2026) extends 2-bit KV cache quantization to autoregressive video generation and shows considerable memory savings with nearly quantitatively lossless performance on video benchmarks, such as VBench (Huang et al., 2024b).

Although previous methods achieve strong video benchmark performance, we discover an overlooked failure of 2-bit KV cache quantization on video world models. As shown in Figure 1, the quantized videos exhibit clear visual degradation compared with BF16, including temporal flickering, blurring, and artifacts, which are not fully captured by video benchmarks even when their reported scores remain nearly unchanged. This unexpected discrepancy motivates us to further investigate where the visual degradation comes from. We first disentangle the effects of Key and Value quantization and find that **the degradation is mainly caused by Keys**, where quantizing Keys alone leads to much worse visual quality than quantizing Values. More unexpectedly, Keys themselves exhibit smaller quantization errors than Values but cause larger output errors. This phenomenon indicates that the sensitivity of Keys cannot be explained by quantization error magnitude alone, where more potential issues are hidden (Tuncer et al., 2026).

To understand this challenge, we revisit the attention computation in Transformers. Keys determine the matching scores between Queries and cached tokens through $QK^\top$. Therefore, **Key perturbations can alter attention logits and change the ranking of temporal-spatial tokens, which may cause frequent token-selection shifts**. In video world models, such shifts will redirect Queries to incorrect cached frames or spatial regions, which leads to temporal flickering and visual degradation.

Based on these insights, we propose **Quant*WM***, a training-free 2-bit KV cache quantization framework that preserves attention behavior during quantization on video world models. Specifically, we first introduce *quantization-sensitivity-aware clustering (QSAC)*, which builds on the centroid-residual quantization scheme of QVG and reduces the impact of Key quantization errors on attention logits. Instead of selecting centroids according to K-means clustering (McQueen, 1967), QSAC jointly considers Query sensitivity and the dynamic range of the K residuals to select centroids with smaller INT2 residual quantization impact on $QK^\top$. Secondly, to further correct the attention shifts, we introduce *principal-subspace attention compensation (PSAC)* to directly compensate the attention logits during inference. In detail, PSAC extracts the dominant Query subspace and represents the remaining Key error components along these sensitive directions with low-rank projections.

We conduct extensive experiments on diverse representative video world models, including LingBot-World-v2 (Gao et al., 2026), Matrix-Game-2 (He et al., 2025) and HY-World 1.5 (Sun et al., 2025), and two video generation models LongCat-Video (Team et al., 2025) and Causal-Forcing (Zhu et al.,

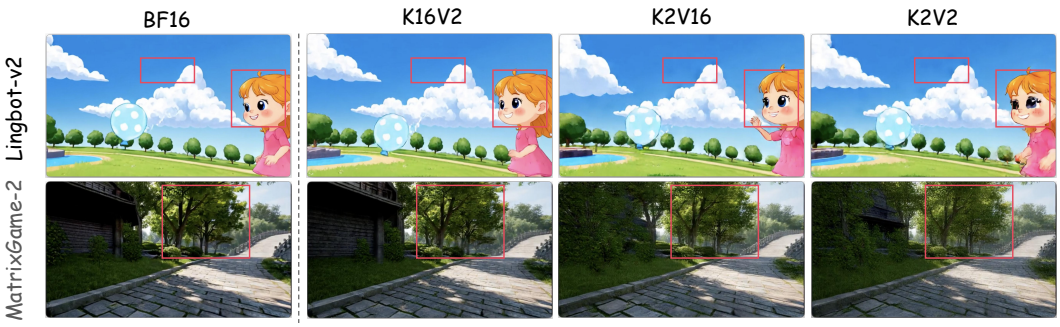


Figure 2: Visual comparison of BF16, K16V2, K2V16 and K2V2 on video world models. As indicated in the red bounding box area, K16V2 preserves the visual quality well, but K2V16 introduces temporal flickering (above) and visual degradation (below), which indicates Key is more sensitive and dominates the performance degradation.

2026). The results show that **Quant*WM*** consistently alleviates the temporal flickering and visual degradation, while maintaining strong performance on benchmarks. Meanwhile, **Quant*WM*** achieves up to $6.20\times$ KV cache memory compression with only limited inference overhead, which demonstrates its effectiveness and efficiency for KV cache quantization on video world models.

2 Preliminary

Quantization is a popular model compression technique which can convert values from full precision to low-bit representations. Given a tensor X, its b-bit affine quantization and dequantization is formulated as:

$$X_q = \text{clip}\left(\left\lfloor \frac{X}{s} \right\rceil + z, q_{\min}, q_{\max}\right), \qquad \hat{X} = \mathcal{Q}_b(X) = s(X_q - z), \tag{1}$$

where s and z denote the scaling factor and zero-point, respectively, which can be elaborated as:

$$s = \frac{X_{\max} - X_{\min}}{q_{\max} - q_{\min}}, \qquad z = \left\lfloor q_{\min} - \frac{X_{\min}}{s} \right\rceil. \tag{2}$$

The quantization parameters can be shared at different granularities, such as per-token, per-channel, or per-group quantization. For KV cache quantization, we follow the centroid-residual representation proposed by QVG (Xi et al., 2026). Given a Key token $\mathbf{k}_i \in \mathbb{R}^d$, it is assigned to a centroid $\boldsymbol{\mu}_{z_i}$ through K-means clustering (McQueen, 1967) and decomposed as:

$$\mathbf{r}_i = \mathbf{k}_i - \boldsymbol{\mu}_{z_i}. \tag{3}$$

Then the centroid is preserved at BF16 and the residual tensor $\mathbf{r}_i$ is quantized to low-bit, where the reconstructed Key and its quantization error is defined as:

$$\hat{\mathbf{k}}_i = \boldsymbol{\mu}_{z_i} + \mathcal{Q}_b(\mathbf{r}_i). \tag{4}$$

3 Motivation

3.1 The Overlooked Visual Degradation under 2-Bit KV Quantization

Existing 2-bit KV cache quantization methods report nearly lossless performance on video benchmarks such as VBench. However, when applied to video world models, we find that these metrics do not fully reflect the degradation introduced by aggressive KV quantization. As shown in Figure 1, compared with the BF16 baseline, the state-of-the-art method QVG produces noticeable temporal flickering, blurring and visual artifacts. These observations suggest that video benchmarks alone may overlook important visual degradation introduced by KV cache quantization. Therefore, we further investigate which component of KV cache is mainly responsible for this degradation.

3.2 Key Quantization Dominates Visual Degradation

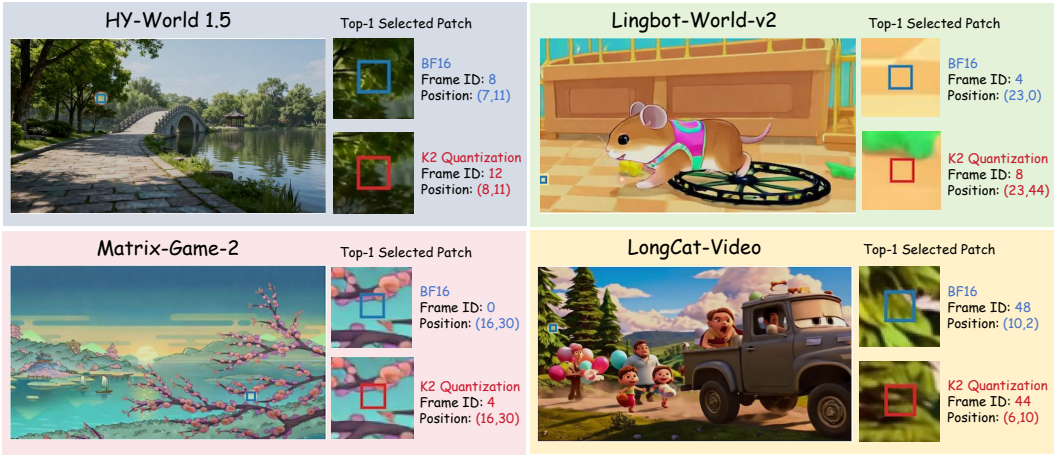


Figure 3: Visualizations of top-1 temporal-spatial token (patch) selection shifts after K quantization under the same query on different video generation and world models.

Table 1: Quantization error of K2/V2-only and their outputs. Key quantization exhibits smaller reconstruction errors but causes larger output degradation than Value quantization.

Layer	Model	K2 MSE ↓	V2 MSE ↓	K2-Output MSE ↓	V2-Output MSE ↓
Middle	Matrix-Game-2	**0.1460**	0.2335	0.0853	**0.0560**
	LingBot-v2	**0.2717**	0.3557	0.1868	0.1946
	HY-World 1.5	**0.1340**	0.2973	0.0821	**0.0536**
Last	Matrix-Game-2	**0.0514**	0.3005	0.3107	**0.1860**
	LingBot-v2	**0.1608**	0.3008	0.2256	**0.1039**
	HY-World 1.5	**0.0090**	0.2939	0.5884	**0.1679**

To locate the main source of visual degradation, we separately quantize Key and Value caches using QVG. As shown in Figure 2, K2V16 produces more severe visual degradation than K16V2. On these samples, the same trend is reflected by frame-level quality metrics as shown in Table 2, where K16V2 achieves clearly better PSNR, SSIM and LPIPS than K2V16, while K2V2 performs similarly to K2V16. These results indicate that the degradation of 2-bit KV cache quantization on video world models is mainly dominated by Key quantization.

Table 2: Visual quality of the samples under different setting, where K2-only causes larger visual degradation than V2-only. MG-2 and LB-2 are short for Matrix-Game-2 and Lingbot-World-v2, respectively.

Model	Setting	PSNR ↑	SSIM ↑	LPIPS ↓
MG-2	K2V16	18.47	0.547	0.219
	K16V2	**21.58**	**0.708**	**0.126**
	K2V2	18.32	0.500	0.261
LB-2	K2V16	15.38	0.355	0.263
	K16V2	**17.70**	**0.476**	**0.157**
	K2V2	15.79	0.376	0.264

More surprisingly, as shown in Table 1, the reconstruction error of quantized Keys is even smaller than that of Values, while K2V16 generally introduces a larger output error than K16V2. This suggests that the sensitivity of Keys cannot be explained by the magnitude of their quantization error.

3.3 Key Quantization Perturbs Temporal-Spatial Token Selection

To understand why Key is quantization-sensitive, we revisit the attention computation. Given Query $\mathbf{Q}$ and Key $\mathbf{K}$, the attention logits and the post-quantization logits perturbation can be elaborated as:

$$\mathbf{L} = \frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}, \qquad \Delta\mathbf{L} = \frac{\mathbf{Q}(\mathbf{K} - \hat{\mathbf{K}})^\top}{\sqrt{d}}. \tag{5}$$

Therefore, even a small Key perturbation may change the relative ordering of attention logits and shift the temporal-spatial tokens selected by Queries.

To validate this analysis, we visualize the temporal-spatial tokens selected by Queries. As shown in Figure 3, 2-bit Key causes clear token-selection shifts compared with BF16, where Queries attend to different historical frames or spatial regions. In video world models, such attention mismatch may redirect the current chunk to inconsistent historical frames or spatial regions, which finally leads to temporal flickering and visual degradation. Accordingly, in addition to reconstruction error, KV cache quantization should also consider to reduce its impact on attention logits.

4 QUANT*WM*

In this section, we present **Quant*WM***, a training-free 2-bit KV cache quantization framework for video world models. We first introduce *quantization-sensitivity-aware clustering (QSAC)* to reduce the impact of Key quantization on attention logits. And then we present *principal-subspace attention compensation (PSAC)* to further correct the attention shifts.

4.1 QUANTIZATION-SENSITIVITY-AWARE CLUSTERING (QSAC)

Existing centroid-residual KV quantization methods (QVG) typically apply K-means to assign each Key token to its nearest centroid and then quantizing the residual tensor. Such strategy mainly considers the distance between tokens and centroids but ignores the impact of the quantized residual tensor. As discussed in Section 3, even a small Key quantization error may cause large perturbations to attention logits when it interacts with sensitive Query channels. To address this issue, we introduce QSAC, which selects centroids based on the impact of residual quantization on attention logits. Specifically, it jointly considers Query-channel sensitivity and the dynamic range of residual groups so that it can introduce smaller quantization perturbations to $QK^\top$.

To approximate the Q-channel sensitivity during causal generation, we use the Queries observed in previous generation steps and maintain their second-order statistics for each attention head h:

$$\mathbf{M}_h^{t-1} = \frac{1}{N_{<t}} \sum_{\tau<t} \sum_i \mathbf{q}_{\tau,i,h} \mathbf{q}_{\tau,i,h}^\top. \tag{6}$$

The second-order Query statistics directly follows from the squared attention-logit error introduced by Key quantization. Given a Key quantization error $\mathbf{e} = \mathbf{k} - \hat{\mathbf{k}}$, we have:

$$\mathbb{E}_{\mathbf{q}} \left[(\mathbf{q}^\top \mathbf{e})^2 \right] = \mathbf{e}^\top \mathbf{M}_h^{t-1} \mathbf{e}, \tag{7}$$

where $\mathbf{M}_h^{t-1}$ characterizes the sensitivity of different Key-error directions to historical Queries. However, directly using the full matrix $\mathbf{M}_h^{t-1}$ for centroid assignment introduces additional computation overheads. To reduce computation, we use its diagonal approximation, where $[\mathbf{M}_h^{t-1}]_{cc} = \mathbb{E}[q_c^2]$ weights the contribution of channel c to the expected logit perturbation:

$$\mathbf{e}^\top \mathbf{M}_h^{t-1} \mathbf{e} \approx \sum_c [\mathbf{M}_h^{t-1}]_{cc} e_c^2, \qquad w_{h,c} = \frac{[\mathbf{M}_h^{t-1}]_{cc}}{\frac{1}{d} \sum_{c'} [\mathbf{M}_h^{t-1}]_{c'c'}}. \tag{8}$$

Channels with larger $w_{h,c}$ are more sensitive to Key quantization errors. Based on this sensitivity, QSAC first defines a Query-aware distance:

$$d_{\mathrm{Q}}(\mathbf{k}_i, \boldsymbol{\mu}_j) = \sum_c w_{h,c} (k_{i,c} - \mu_{j,c})^2, \tag{9}$$

which is used to efficiently select a top-M candidate centroids set $\mathcal{C}_i$. Subsequently, we consider the quantization difficulty of the residuals. For each candidate centroid $\boldsymbol{\mu}_j$, we define the residual and its group-wise dynamic range as:

$$\mathbf{r}_{ij} = \mathbf{k}_i - \boldsymbol{\mu}_j, \qquad R_g(\mathbf{r}_{ij}) = \max_{c \in \mathcal{G}_g} r_{ij,c} - \min_{c \in \mathcal{G}_g} r_{ij,c}, \tag{10}$$

where $\mathcal{G}_g$ denotes the set of channels in the g-th group. For uniform b-bit quantization, the corresponding scaling factor is $\Delta_g = R_g/(2^b - 1)$. Then according to (Widrow et al., 1996), the quantization error can be approximated as:

$$\mathbb{E}[e_c^2] \approx \frac{\Delta_g^2}{12} = \frac{R_g^2}{12(2^b - 1)^2}. \tag{11}$$

Since $1/[12(2^b-1)^2]$ is shared by all candidate centroids, we remove it and combine Eq. 11 with Eq. 8 to get the expected squared attention-logit perturbation for a candidate centroid, which we define it as the QSAC score and then use it to select the optimal centroid:

$$S(\mathbf{k}_i, \boldsymbol{\mu}_j) = \sum_g \left(\sum_{c \in \mathcal{G}_g} w_{h,c} \right) R_g(\mathbf{r}_{ij})^2, \qquad z_i = \arg\min_{j \in \mathcal{C}_i} S(\mathbf{k}_i, \boldsymbol{\mu}_j). \tag{12}$$

In this way, QSAC can prioritizes Query-sensitive groups and successfully incorporates the approximated residual quantization error into centroid selection, so that the selected centroid is expected to introduce smaller perturbations to $QK^\top$ after quantization.

4.2 Principal-Subspace Attention Compensation (PSAC)

Due to the limited representation capacity of 2-bit quantization, Key quantization errors cannot be fully eliminated even after QSAC, which may still cause attention shifts. Therefore, we further introduce PSAC to directly correct the remaining logit perturbations.

Let $\hat{\mathbf{K}}_0$ denote the dequantized Key after QSAC, and we define the remaining quantization error as:

$$\mathbf{E} = \mathbf{K} - \hat{\mathbf{K}}_0. \tag{13}$$

Reusing the historical Query statistics $\mathbf{M}_h^{t-1}$ introduced in Section 4.1, the expected logit perturbations caused by $\mathbf{E}$ can be written as:

$$\mathcal{D}(\mathbf{E}) = \mathbb{E}_{\mathbf{Q}} \left[\left\| \mathbf{Q}\mathbf{E}^\top \right\|_F^2 \right] = \mathrm{Tr}\left(\mathbf{E}\mathbf{M}_h^{t-1}\mathbf{E}^\top \right). \tag{14}$$

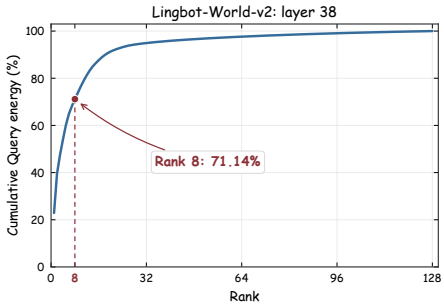


Figure 4: Cumulative Query energy in LingBot-v2, where top-8 directions include most of the total Query energy.

However, directly storing full quantization error $\mathbf{E}$ for logits compensation would introduce additional memory overheads. Inspired by (Liu et al., 2025; Zhao et al., 2025b) which claim that the hidden representations in Transformers often exhibit low-rank structures, we examine the energy distribution of historical Queries and find that the energy is highly concentrated in a few principal directions. As shown in Figure 4, the top-8 directions include more than 60% of the total Query energy, which suggest that we can focus on the error components along these dominant Query directions. Specifically, we perform eigenvalue decomposition for $\mathbf{M}_h^{t-1}$:

$$\mathbf{M}_h^{t-1} = \mathbf{U}_h \mathbf{\Lambda}_h \mathbf{U}_h^\top, \qquad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d. \tag{15}$$

Each eigenvalue represents the historical Query energy along its corresponding eigenvector. Therefore, Key quantization errors projected onto directions with larger eigenvalues have a larger expected impact on attention logits. We preserve the top-r eigenvectors $\mathbf{U}_r$ as the principal Query subspace and restore the remaining Key quantization error along these directions:

$$\mathbf{C} = \mathbf{E}\mathbf{U}_r, \tag{16}$$

where we preserve both $\mathbf{C}$ and $\mathbf{U}_r$ during inference. Finally, we compensate these error components after quantization, where the attention logits will be corrected as follows:

$$\hat{\mathbf{K}} = \hat{\mathbf{K}}_0 + \mathbf{C}\mathbf{U}_r^\top, \qquad \mathbf{Q}\hat{\mathbf{K}}^\top = \mathbf{Q}\hat{\mathbf{K}}_0^\top + \mathbf{Q}\mathbf{U}_r\mathbf{C}^\top. \tag{17}$$

In our implementation, we set $r = 8$. Since both $\mathbf{C}$ and $\mathbf{U}_r$ are low-rank, PSAC can directly compensates the attention-logit shift without extensive extra overheads.

5 Experiments

5.1 Experimental Setup

Models To validate the effectiveness of **Quant*WM***, we conduct extensive experiments on 3 open-sourced video world models, including Matrix-Game-2 (He et al., 2025), HY-World 1.5 (Sun et al., 2025) and LingBot-World-v2 (Gao et al., 2026). We also include 2 autoregressive video generation models LongCat-Video (Team et al., 2025) and Causal-Forcing (Zhu et al., 2026) to evaluate the generalization. We conduct the main evaluation on 480p and 720p generated videos with 93 frames.



Figure 5: Visual quality comparison of **Quant*WM***, QVG and BF16. QVG suffers from temporal flickering and visual degradation, while our **Quant*WM*** preserves clearer details and visual quality.

Table 3: Visual quality and VBench comparison results of **Quant*WM*** and baselines on 480p videos. **Quant*WM*** significantly improves visual quality while maintaining strong VBench performance.

Model	Method	Frame-level Quality			*VBench*					
		PSNR ↑	**SSIM ↑**	**LPIPS ↓**	*Subject ↑*	*Background ↑*	*Aesthetic ↑*	*Image ↑*	*Temporal ↑*	***Rank ↓***
Matrix-Game-2	BF16	∞	1.0000	0.0000	0.9104	0.9423	0.5846	0.6952	0.9553	–
	KIVI	19.298	0.6557	0.1797	0.9097	0.9418	0.5841	0.6914	**0.9603**	1.80
	QVG	17.749	0.5651	0.2851	0.8894	0.9320	0.5831	0.6603	0.9558	3.00
	Ours	**20.615**	**0.7212**	**0.1443**	**0.9109**	**0.9421**	**0.5883**	**0.6933**	0.9561	**1.20**
LingBot-v2	BF16	∞	1.0000	0.0000	0.8877	0.9126	0.5949	0.6962	0.9314	–
	KIVI	14.799	0.5950	0.2508	0.8869	**0.9137**	0.5883	0.6892	0.9320	2.40
	QVG	14.414	0.5726	0.2779	0.8860	0.9127	0.5912	0.6929	**0.9329**	2.20
	Ours	**16.138**	**0.6688**	**0.1893**	**0.8894**	0.9134	**0.5947**	**0.6953**	0.9323	**1.40**
HY-World 1.5	BF16	∞	1.0000	0.0000	0.9588	0.9520	0.6378	0.7209	0.9562	–
	KIVI	17.677	0.7165	0.1540	0.9520	0.9494	0.6362	0.7049	**0.9702**	2.20
	QVG	16.118	0.6559	0.2114	0.9503	0.9488	**0.6394**	0.7122	0.9585	2.20
	Ours	**18.449**	**0.7661**	**0.1082**	**0.9600**	**0.9528**	0.6366	**0.7196**	0.9584	**1.60**
LongCat-Video	BF16	∞	1.0000	0.0000	0.9747	0.9605	0.6148	0.6993	0.9461	–
	KIVI	24.448	0.9180	0.0399	**0.9754**	0.9610	0.6141	0.6695	0.9475	2.20
	QVG	20.683	0.8556	0.0784	0.9739	0.9603	0.6144	0.7001	0.9463	2.60
	Ours	**25.508**	**0.9286**	**0.0332**	0.9748	**0.9615**	**0.6154**	**0.7002**	**0.9489**	**1.20**
Causal-Forcing	BF16	∞	1.0000	0.0000	0.9327	0.9421	0.6348	0.7028	0.9534	–
	KIVI	15.585	0.7201	0.1775	0.9334	0.9426	0.6343	0.7034	0.9540	2.20
	QVG	14.887	0.6773	0.2185	0.9311	0.9412	0.6317	0.6993	0.9546	2.80
	Ours	**16.283**	**0.7464**	**0.1554**	**0.9342**	**0.9427**	**0.6354**	**0.7044**	**0.9558**	**1.00**

Evaluations We select Quant VideoGen (Xi et al., 2026), the most relevant 2-bit KV cache quantization method for video generation, as the main baseline. The representative LLMs quantization method KIVI (Liu et al., 2024b) is also included into comparison. We mainly evaluate the visual degradation introduced by quantization using frame-level visual quality and qualitative visualizations. During experiments, we compute them over all frames of each video. We also report performance on VBench (Huang et al., 2024b). Although it may not fully capture the visual degradation studied in this work, it is still an important benchmark for evaluating the overall quality of generated videos. Following QVG, we use the prompt suite from MovieGen Benchmark (Polyak et al., 2024).[1] For Matrix-Game-2, we use the official first frames combined with 7–8 different action sequences as inputs.[2]. For fair comparison, we reproduce all the baselines with their official repository, follow the native KV cache convention of each model and apply all methods to the same cached tensors.

Implementation We implement **Quant*WM*** in PyTorch and conduct all experiments on NVIDIA A800 80GB GPUs. Following QVG, we apply streaming chunk-wise KV cache quantization. We use 256 centroids and quantization groupsize 64 (QVG with 64 and KIVI with 32), with BF16

[1] `facebookresearch/MovieGenBench`
[2] `SkyworkAI/Matrix-Game/tree/main/Matrix-Game-2/demo_images`

Table 4: Visual quality and VBench comparison results of **Quant*WM*** and baselines on 720p videos.

Model	Method	Frame-level Quality			*VBench*					
		PSNR ↑	**SSIM ↑**	**LPIPS ↓**	*Subject ↑*	*Background ↑*	*Aesthetic ↑*	*Image ↑*	*Temporal ↑*	***Rank ↓***
Matrix-Game-2	BF16	∞	1.0000	0.0000	0.9096	0.9490	0.5875	0.7038	0.9550	–
	KIVI	19.200	0.6659	0.2048	0.9081	0.9479	0.5888	0.7003	**0.9601**	1.80
	QVG	17.721	0.5871	0.3047	0.8884	0.9373	0.5840	0.6679	0.9556	3.00
	Ours	**20.568**	**0.7234**	**0.1651**	**0.9100**	**0.9488**	**0.5914**	**0.7020**	0.9559	**1.20**
LingBot-v2	BF16	∞	1.0000	0.0000	0.8984	0.9202	0.6224	0.7211	0.9362	–
	KIVI	15.134	0.6317	0.2520	0.8965	**0.9224**	0.6180	0.7165	0.9364	2.40
	QVG	13.927	0.5958	0.3220	0.8906	0.9195	0.6181	0.7175	**0.9367**	2.20
	Ours	**15.564**	**0.6540**	**0.2267**	**0.8994**	0.9213	**0.6214**	**0.7207**	0.9365	**1.40**
HY-World 1.5	BF16	∞	1.0000	0.0000	0.9592	0.9629	0.6556	0.7236	0.9659	–
	KIVI	17.643	0.7338	0.1747	0.9523	0.9593	0.6518	0.7119	**0.9699**	2.20
	QVG	16.085	0.6810	0.2358	0.9505	0.9583	**0.6575**	0.7181	0.9678	2.20
	Ours	**18.423**	**0.7779**	**0.1257**	**0.9602**	**0.9636**	0.6551	**0.7229**	0.9667	**1.60**
LongCat-Video	BF16	∞	1.0000	0.0000	0.9717	0.9678	0.6327	0.6975	0.9609	–
	KIVI	22.896	0.8805	0.0708	**0.9719**	**0.9675**	0.6334	0.6962	0.9610	1.90
	QVG	20.988	0.8246	0.1098	0.9714	0.9671	0.6325	0.6963	0.9609	2.80
	Ours	**25.127**	**0.9094**	**0.0503**	0.9715	**0.9675**	**0.6340**	**0.6978**	**0.9620**	**1.30**
Causal-Forcing	BF16	∞	1.0000	0.0000	0.9345	0.9514	0.6340	0.7038	0.9525	–
	KIVI	15.394	0.7349	0.2013	0.9335	0.9517	0.6349	0.7023	0.9532	2.20
	QVG	14.771	0.6996	0.2390	0.9321	0.9498	0.6321	0.6982	0.9543	2.80
	Ours	**16.231**	**0.7628**	**0.1716**	**0.9354**	**0.9521**	**0.6352**	**0.7034**	**0.9554**	**1.00**

Table 5: Top-1 temporal-spatial token selection shift ratio on 480p videos. **Quant*WM*** significantly reduces the token-selection shifts caused by 2-bit Key quantization.

Method	Matrix-Game-2	LingBot-v2	HY-World 1.5	LongCat-Video	Causal-Forcing
BF16	0.0000	0.0000	0.0000	0.0000	0.0000
QVG	0.5388	0.5491	0.6136	0.5034	0.3728
Ours	**0.1019**	**0.1600**	**0.1392**	**0.2225**	**0.1420**

centroids and INT2 quantization for the residuals. For PSAC, it extracts the top-8 eigenvectors $\mathbf{U}_r$ of $\mathbf{M}_h^{t-1}$ as the principal Query subspace. We store $\mathbf{U}_r$ in BF16 and quantize $\mathbf{C} = \mathbf{E}\mathbf{U}_r$ to INT8.

5.2 Main Results and Visualizations

Visual quality evaluation As shown in Table 3, 4 and Figure 5, **Quant*WM*** consistently improves frame-level visual quality over QVG across all five models while visibly reducing temporal flickering, blurring and artifacts. For example, on LongCat-Video, PSNR improves from 20.683 to 25.508 and LPIPS decreases from 0.0784 to 0.0332. Although KIVI benefits from a finer groupsize and a BF16 recent-window cache, **Quant*WM*** still achieves the best results across all models. Long video generation results and more visualizations are provided in Appendix B.

Video benchmark evaluation **Quant*WM*** also maintains strong performance and the best average rank on VBench across all five models. Notably, the Temporal scores are very close among different methods, even when QVG exhibits clear temporal flickering in Figure 5, indicating that this metric does not fully capture such visual degradation. Considering both metrics, **Quant*WM*** demonstrates stronger ability to preserve visual quality under 2-bit KV cache quantization on video world models

5.3 Temporal-spatial token selection shifts ratio

To further verify whether **Quant*WM*** preserves attention logits under 2-bit quantization, we evaluate the top-1 temporal-spatial token selection shift ratio compared with BF16. As shown in Table 5, QVG causes frequent token-selection changes across all five models, with shift ratios up to 61.36%. In contrast, reduces the average shift ratio from 51.55% to 15.31%. These results further proves that our **Quant*WM*** successfully reduces the quantization perturbation to the attention.

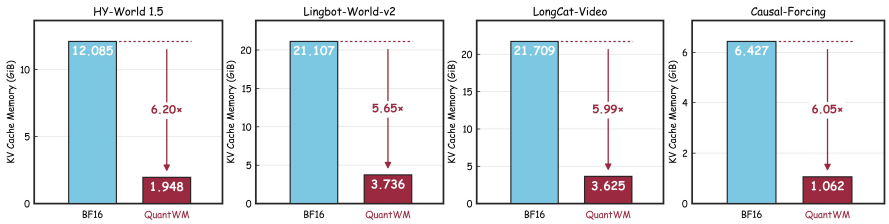


Figure 6: KV cache memory comparison of BF16 and **Quant*WM*** for 93-frame video generation. **Quant*WM*** achieves up to $6.20\times$ KV cache memory compression across different models.

5.4 Ablation Study

To demonstrate the contributions of QSAC and PSAC, We conduct an ablation study on HY-World 1.5. As shown in Table 6, QSAC consistently improves frame-level quality metrics while reducing the token shift ratio from 61.36% to 38.17%. This demonstrates that selecting centroids according to Query sensitivity and residual quantization difficulty effectively reduces harmful Key quantization perturbations. PSAC further shows a stronger effect on attention preservation, where it reduces the token Shift to 22.9% even without QSAC. Finally, combining QSAC and PSAC achieves the best overall performance.

Table 6: Ablation study of **Quant*WM*** on HY-World 1.5.

QSAC	PSAC	PSNR ↑	SSIM ↑	LPIPS ↓	Token Shift ↓
×	×	16.118	0.6559	0.2114	0.6136
✓	×	17.990	0.7423	0.1285	0.3817
×	✓	17.849	0.7358	0.1312	0.2290
✓	✓	**18.449**	**0.7661**	**0.1082**	**0.1392**

5.5 Inference Efficiency

KV cache memory usage We first evaluate the KV cache memory consumption during 93-frame video generation. As shown in Figure 6, **Quant*WM*** consistently reduces the KV cache footprint across different models, achieving up to $6.20\times$ compression. For example, on LongCat-Video, **Quant*WM*** reduces the KV cache memory from 21.709 GiB to 3.625 GiB. These results demonstrate the effectiveness of **Quant*WM*** in reducing the memory cost during generation.

End-to-end latency As shown in Table 7, **Quant*WM*** introduces only limited latency overhead across different models. The generation latency increases by only 5.44% on LingBot-World-v2 and 5.87% on HY-World 1.5, while it is reduced by 17.78% on LongCat-Video. These results indicate that **Quant*WM*** achieves significant memory compression with limited impact on end-to-end generation latency. Detailed implementations and analysis of real-quant inference system design are provided in Appendix D.

Table 7: Inference latency comparison between BF16 and **Quant*WM***.

Model	BF16 (s)	Quant*WM* (s)
LingBot-World-v2	193.56	204.09 5.44%↑
HY-World 1.5	112.17	118.75 5.87%↑
LongCat-Video	255.35	209.94 **17.78%**↓

6 Conclusion

In this paper, we investigate the overlooked visual degradation caused by KV cache quantization in video world models. We show that Key quantization is more sensitive than Value because Key perturbations can influence attention logits and shift temporal-spatial token selection. Based on this observation, we propose **Quant*WM***, a training-free 2-bit KV cache quantization framework that considers attention preservation. **Quant*WM*** includes quantization-sensitivity-aware clustering (QSAC), which incorporates Query sensitivity and residual quantization difficulty into centroid selection, and principal-subspace attention compensation (PSAC), which directly corrects the remaining logit perturbations along dominant Query directions. Experiments across multiple video world models demonstrate that **Quant*WM*** effectively alleviates temporal flickering and visual degradation, while achieving high KV cache memory compression with limited overheads.

AI USE STATEMENT

In this work, we used generative AI tools for retrieving literature and polishing context. We have not used generative AI tools for developing methods or writing codes, and proving mathematical claims and generating synthetic datasets are not applicable to this work. We have reviewed all AI-assisted work. We have checked LLM-retrieved literature and LLM-polished context by 3 authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies efficient KV cache quantization for video world models and does not introduce new generative capabilities or involve human subjects. **Quant*WM*** inherits the potential risks of the underlying generative models, including possible misuse for generating misleading or inappropriate visual content. Since improved memory efficiency may lower the hardware barrier for deployment, we encourage responsible use of the proposed method in accordance with the licenses and usage policies of the underlying models.

REPRODUCIBILITY STATEMENT

All experiments in this work are conducted on publicly available video world models using open-source evaluation data and benchmarks. We provide detailed descriptions of the model settings, quantization configurations, evaluation metrics and implementation details in the main paper and Appendix to facilitate reproduction. We promise to release the codes of **Quant*WM*** and the project page as soon as possible.

REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. *Advances in Neural Information Processing Systems*, 37:58757–58791, 2024.

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

Zelin Gao, Qiuyu Wang, Jiapeng Zhu, Jingye Chen, Zichen Liu, Qingyan Bai, Jiahao Wang, Yufeng Yuan, Hanlin Wang, Yichong Lu, et al. Infinite worlds with versatile interactions. *arXiv preprint arXiv:2607.07534*, 2026.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, et al. Matrix-game 2.0: An open-source real-time and streaming interactive world model. *arXiv preprint arXiv:2508.13009*, 2025.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Yakun S Shao, Kurt Keutzer, and Amir Gholami. Kvquant: Towards 10 million context length llm inference with kv cache quantization. *Advances in Neural Information Processing Systems*, 37:1270–1303, 2024.

Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and Xiaojuan Qi. Slim-llm: Salience-driven mixed-precision quantization for large language models. *arXiv preprint arXiv:2405.14917*, 2024a.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 38:167283–167308, 2026.

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. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024b.

Junyan Li, Yang Zhang, Muhammad Yusuf Hassan, Talha Chafekar, Tianle Cai, Zhile Ren, Pengsheng Guo, Foroozan Karimzadeh, Chong Wang, and Chuang Gan. Commvq: Commutative vector quantization for kv cache compression. *arXiv preprint arXiv:2506.18879*, 2025.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems*, 6:87–100, 2024.

Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. Qserve: W4a8kv4 quantization and system co-design for efficient llm serving. *Proceedings of Machine Learning and Systems*, 7, 2025.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. In *International Conference on Learning Representations*, volume 2026, pp. 91177–91196, 2026.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. Llm-qat: Data-free quantization aware training for large language models. In *Findings of the association for computational linguistics: ACL 2024*, pp. 467–484, 2024a.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750*, 2024b.

Ziyue Liu, Ruijie Zhang, Zhengyang Wang, Mingsong Yan, Zi Yang, Paul D Hovland, Bogdan Nicolae, Franck Cappello, Sui Tang, and Zheng Zhang. Cola: Compute-efficient pre-training of llms via low-rank activation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4627–4645, 2025.

James B McQueen. Some methods of classification and analysis of multivariate observations. In *Proc. of 5th berkeley symposium on math. stat. and prob.*, pp. 281–297, 1967.

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *International conference on machine learning*, pp. 7197–7206. PMLR, 2020.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Gao Peng, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 45472–45496, 2024.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

Wenqiang Sun, Haiyu Zhang, Haoyuan Wang, Junta Wu, Zehan Wang, Zhenwei Wang, Yunhong Wang, Jun Zhang, Tengfei Wang, and Chunchao Guo. Worldplay: Towards long-term geometric consistency for real-time interactive world modeling. *arXiv preprint arXiv:2512.14614*, 2025.

Meituan LongCat Team, Bairui Wang, Bin Xiao, Bo Zhang, Bolin Rong, Borun Chen, Chang Wan, Chao Zhang, Chen Huang, Chen Chen, et al. Longcat-flash-omni technical report. *arXiv preprint arXiv:2511.00279*, 2025.

Tuna Tuncer, Felix Becker, and Thomas Pfeil. Quantized keys steal attention: Bias correction for kv-cache compression in video diffusion. *arXiv preprint arXiv:2605.26266*, 2026.

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion models are real-time game engines. In *International Conference on Learning Representations*, volume 2025, pp. 73754–73776, 2025.

Zile Wang, Zexiang Liu, Jiaxing Li, Kaichen Huang, Baixin Xu, Fei Kang, Mengyin An, Peiyu Wang, Biao Jiang, Yichen Wei, et al. Matrix-game 3.0: Real-time and streaming interactive world model with long-horizon memory. *arXiv preprint arXiv:2604.08995*, 2026.

Dirk Weissenborn, Oscar Täckström, and Jakob Uszkoreit. Scaling autoregressive video models. *arXiv preprint arXiv:1906.02634*, 2019.

Bernard Widrow, Istvan Kollar, and Ming-Chang Liu. Statistical theory of quantization. *IEEE Transactions on instrumentation and measurement*, 45(2):353–361, 1996.

Haocheng Xi, Shuo Yang, Yilong Zhao, Muyang Li, Han Cai, Xingyang Li, Yujun Lin, Zhuoyang Zhang, Jintao Zhang, Xiuyu Li, et al. Quant videogen: Auto-regressive long video generation via 2-bit kv-cache quantization. *arXiv preprint arXiv:2602.02958*, 2026.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International conference on machine learning*, pp. 38087–38099. PMLR, 2023.

Zeqi Xiao, Yushi Lan, Yifan Zhou, Wenqi Ouyang, Shuai Yang, Yanhong Zeng, and Xingang Pan. Worldmem: Long-term consistent world simulation with memory. *Advances in Neural Information Processing Systems*, 38:49632–49652, 2026.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025a.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025b.

Tianyi Zhang, Jonah Yi, Zhaozhuo Xu, and Anshumali Shrivastava. Kv cache is 1 bit per channel: Efficient large language model inference with coupled quantization. *Advances in Neural Information Processing Systems*, 37:3304–3331, 2024.

Jiaqi Zhao, Miao Zhang, Chao Zeng, Ming Wang, Xuebo Liu, and Liqiang Nie. Lrquant: Learnable and robust post-training quantization for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2240–2255, 2024.

Jiaqi Zhao, Chao Zeng, Ming Wang, Linxuan Han, Yuzhang Shang, Miao Zhang, and Liqiang Nie. Lrquant: A unified and learnable framework to post-training quantization for transformer-based large foundation models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025a.

Jiaqi Zhao, Miao Zhang, Deng Xiang, Ming Li, Weili Guan, and Liqiang Nie. Boost post-training quantization via null space optimization for large language models. *arXiv preprint arXiv:2506.11044*, 2025b.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time interactive video generation. *arXiv preprint arXiv:2602.02214*, 2026.

A Related Works

A.1 Video Generation and World Models

Diffusion-based video generation has advanced rapidly through large-scale spatial-temporal Transformers, which significantly improves visual fidelity and temporal consistency (Singer et al., 2022; Zheng et al., 2024; Yang et al., 2025b). Recent autoregressive approaches further enable long-form and streaming generation by producing videos chunk by chunk under causal attention (Huang et al., 2026; Liu et al., 2026; Zhu et al., 2026; Yang et al., 2025a). Building on these advances, video world models incorporate action and camera controls to support real-time interaction with generated environments (Alonso et al., 2024; He et al., 2025; Gao et al., 2026; Valevski et al., 2025). Long-term consistency has also motivated explicit memory mechanisms that retrieve or reconstruct relevant historical observations (Xiao et al., 2026; Sun et al., 2025; Wang et al., 2026). Nevertheless, causal video models commonly storage historical representations through KV caches. Since each generated chunk introduces numerous spatial-temporal tokens, the cache grows rapidly with the cached context and will finally dominate GPU memory, making KV cache efficiency a central bottleneck for video generation and world modeling.

A.2 Model Quantization

Quantization (Nagel et al., 2020; Liu et al., 2024a) reduces storage and inference overheads by transfer high-precision tensors to low-bit values. Post-training quantization (Shao et al., 2024; Huang et al., 2024a; Zhao et al., 2025a) has become particularly attractive because it avoids costly model retraining. Representative methods include GPTQ (Frantar et al., 2022), which uses approximate second-order information for weight reconstruction, SmoothQuant (Xiao et al., 2023), which migrates activation outliers into weights for W8A8 quantization, and AWQ (Lin et al., 2024), which protects activation-salient weight channels. These methods primarily focus on efficient inference by compressing model weights and activations.

KV cache quantization further reduces the memory accumulated during autoregressive inference. KIVI (Liu et al., 2024b) applies asymmetric 2-bit quantization to Keys and Values, while KVQuant (Hooper et al., 2024) introduces pre-RoPE Key quantization, non-uniform datatypes, and outlier isolation. CommVQ (Li et al., 2025) instead compresses KV caches using RoPE-compatible learned codebooks. These methods mainly target language models. More recently, Quant-VideoGen (QVG) (Xi et al., 2026) extends 2-bit KV cache quantization to autoregressive video diffusion through semantic-aware smoothing and residual quantization, which achieves nearly lossless performance on video benchmarks. However, we find that such benchmark-level results ignores severe temporal flickering and visual artifacts. Recent work (Tuncer et al., 2026) discovers a systematic softmax attention bias introduced by quantized Keys and corrects it at the attention-score level. In contrast, we focus on quantization perturbations of $QK^\top$ and temporal-spatial token selection, and address them through Query-sensitive centroid assignment and principal-subspace error compensation to improve temporal consistency.

B More Evaluation Results

B.1 Results on Long-video Generation

We further evaluate **Quant*WM*** under 1-minute long-horizon video generation. Long-form autoregressive generation is particularly challenging for low-bit KV cache quantization, since quantization perturbations can accumulate over successive generation steps and lead to larger deviations from the BF16 trajectory. Therefore, we focus on the overall video quality measured by VBench in this setting. As shown in Table 8, **Quant*WM*** outperforms QVG on most VBench dimensions and achieves the best average rank across all five models. For example, on HY-World 1.5, the Image score improves from 0.4756 to 0.6242. Overall, **Quant*WM*** maintains strong video quality under longer generation, demonstrating better robustness to accumulated quantization errors.

Table 8: VBench results on 1-minute video generation. **Quant*WM*** consistently improves video quality over QVG under long-video generation.

Model	Method	*VBench*					
		Subject ↑	*Background* ↑	*Aesthetic* ↑	*Image* ↑	*Temporal* ↑	***Rank*** ↓
Matrix-Game-2	BF16	0.7287	0.8659	0.4475	0.6767	0.9857	–
	QVG	0.7280	0.8675	0.4439	0.5275	0.9858	2.00
	Ours	**0.7328**	**0.8693**	**0.4523**	**0.6639**	**0.9878**	**1.00**
LingBot-v2	BF16	0.9063	0.9318	0.6614	0.7593	0.9786	–
	QVG	0.8946	0.9280	0.6404	0.7620	**0.9774**	1.80
	Ours	**0.9121**	**0.9353**	**0.6567**	**0.7633**	0.9767	**1.20**
HY-World 1.5	BF16	0.7051	0.8242	0.4730	0.6247	0.9656	–
	QVG	0.6591	0.8025	0.4423	0.4756	**0.9795**	1.80
	Ours	**0.7100**	**0.8269**	**0.4895**	**0.6242**	0.9715	**1.20**
LongCat-Video	BF16	0.8340	0.8939	0.6334	0.7396	0.9593	–
	QVG	0.8253	0.8920	0.6222	0.7323	0.9562	2.00
	Ours	**0.8378**	**0.8957**	**0.6443**	**0.7359**	**0.9592**	**1.00**
Causal-Forcing	BF16	0.6903	0.8445	0.4596	0.5647	0.9605	–
	QVG	0.7052	**0.8213**	0.5032	0.6005	0.9497	1.80
	Ours	**0.7184**	0.8155	**0.5070**	**0.6566**	**0.9614**	**1.20**

B.2 More Visualizations of Generated Videos

We provide additional qualitative comparisons across different video generation and world models. As shown in Figure 7, QVG frequently introduces temporal flickering, blurring, and local visual artifacts after 2-bit KV cache quantization. In contrast, **Quant*WM*** better preserves object appearance, spatial details, and temporal consistency, which produces results that remain closer to BF16. These examples further confirm that the temporal consistency improvements of **Quant*WM*** are consistent across different models and generation scenarios.

C Analysis and Illustrations on QSAC and PSAC

C.1 Query Energy Distribution on More Models

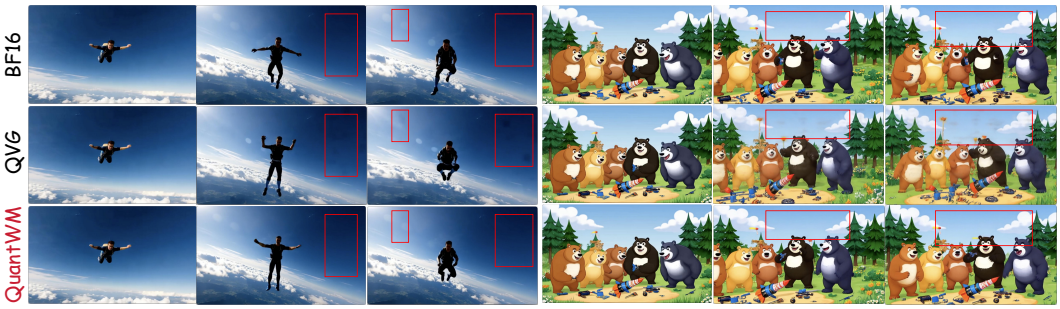
To further validate the low-rank structure of dominant Query directions in PSAC, we visualize the cumulative energy of historical Queries on more video world models. As shown in Figure 8, the top-8 directions capture a large fraction of the total Query energy across different models and layers, which provides a compact principal Query subspace for correcting the Key quantization errors that have larger impacts on attention logits.

C.2 The Effect of the Hyper-parameter r in PSAC

We further study the effect of the PSAC rank r. As shown in Table 9, increasing r generally improves the frame-level quality, but the gains gradually diminish. For example, increasing the rank from 8 to 16 only brings modest improvements in PSNR, SSIM, and LPIPS, while introducing an additional 0.1341 bits on HY-World 1.5 and 0.2569 bits on Causal-Forcing relative to $r = 8$. This trend is also consistent with the cumulative Query-energy distributions in Figure 4 and 8, where the additional energy captured by higher-rank directions gradually decreases. Therefore, we set $r = 8$ throughout our experiments as a practical trade-off between visual quality and additional storage overhead.

C.3 Initialization of Historical Query Statistics

At the beginning of generation, historical Query statistics are not available. To handle this cold-start stage without introducing unreliable estimates, we adopt a simple causal initialization strategy. For



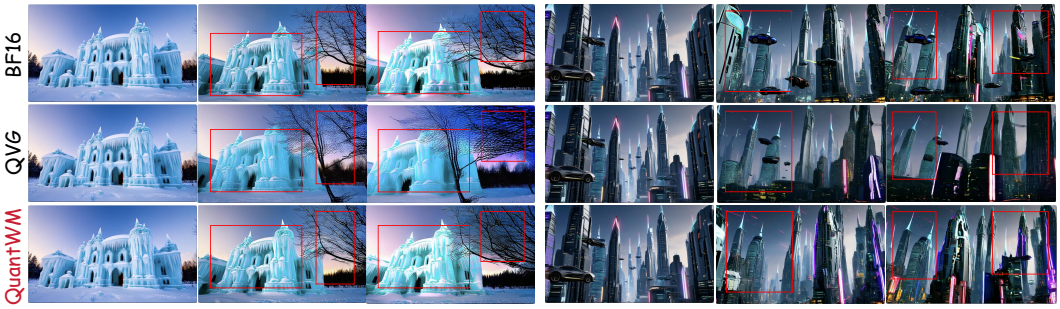
(a) LingBot-World-v2



(b) HY-World 1.5



(c) Matrix-Game-2



(d) Causal-Forcing

Figure 7: More visual comparisons of BF16, QVG and **Quant*WM*** across different video generation and world models. Please refer to Supplement Materials for detailed video comparison.

the first cache chunk, QSAC assigns uniform channel sensitivity, i.e., $w_{h,c} = 1$, such that centroid selection depends only on the residual quantization characteristics, while PSAC is temporarily inactive because no reliable principal Query subspace can be approximated. After the current chunk is quantized, its Query observations are incorporated into the historical second-order statistics and be-

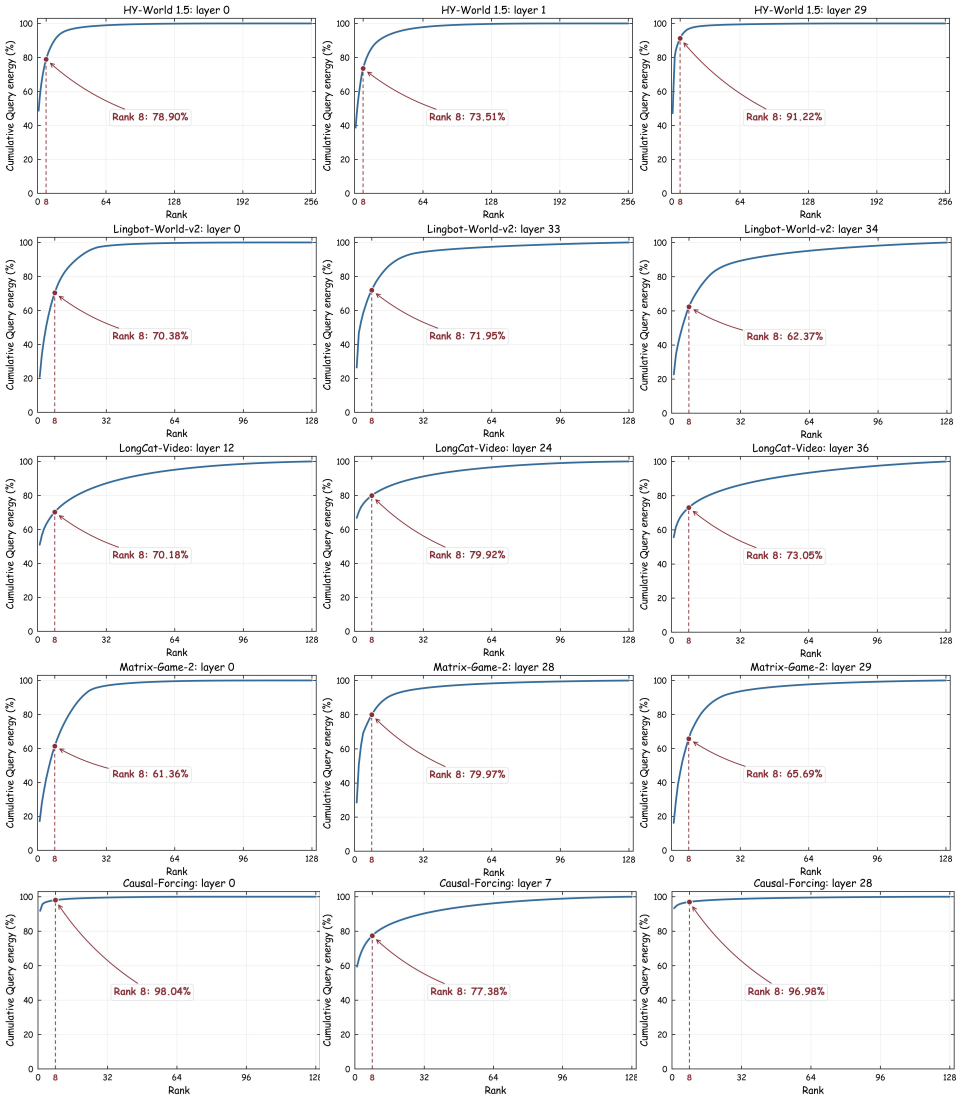


Figure 8: Cumulative Query energy across different models and layers.

come available to subsequent chunks. In this way, the statistics used by QSAC and PSAC are always constructed from previously generated content, which preserve strict causality throughout inference. Since each committed chunk contributes a large number of spatial Query tokens, the historical statistics can be established rapidly after initialization without requiring an additional calibration stage or manually designed warm-up schedule.

C.4 The Effect of the Diagonal Query Approximation in QSAC

QSAC maintains the historical Query second-moment matrix $\mathbf{M}_h^{t-1}$ in Eq. 6, and uses its diagonal entries to construct the Query-aware distance in Eq. 9. To study the effect of the ignored cross-channel correlations, we replace Eq. 9 with the full quadratic form:

$$d_{\text{full}}(\mathbf{k}_i, \boldsymbol{\mu}_j) = (\mathbf{k}_i - \boldsymbol{\mu}_j)^\top \mathbf{M}_h^{t-1} (\mathbf{k}_i - \boldsymbol{\mu}_j), \tag{18}$$

which is used to select the top-m candidate centroids and the remaining QuantWM pipeline is unchanged. As shown in Table 10, retaining the full matrix does not provide consistent improvements in frame-level quality. The differences in PSNR, SSIM and LPIPS are marginal on both HY-World 1.5 and Causal-Forcing, while the full matrix introduces additional inference latency. Therefore, we adopt the diagonal approximation in QSAC throughout our experiments.

Table 9: Ablation study on the rank r of PSAC. Increasing the rank provides marginal quality improvements but introduces higher storage overhead, so we use $r = 8$ to balance visual quality and KV cache compression.

Model	Rank	PSNR ↑	SSIM ↑	LPIPS ↓	Δ Bit vs. $r = 8$
HY-World 1.5	2	18.3949	0.7643	0.1102	−0.1006
	4	18.3973	0.7645	0.1100	−0.0671
	8	18.4485	0.7661	0.1082	0.0000
	16	**18.5368**	0.7692	**0.1063**	+0.1341
	32	18.5157	**0.7694**	0.1067	+0.4023
Causal-Forcing	2	16.2259	0.7457	0.1569	−0.1926
	4	16.2409	0.7466	0.1554	−0.1284
	8	16.2828	0.7464	0.1554	0.0000
	16	**16.3495**	0.7491	0.1527	+0.2569
	32	16.3462	**0.7507**	**0.1515**	+0.7706

Table 10: Effect of the diagonal approximation in QSAC.

Model	QSAC Metric	PSNR ↑	SSIM ↑	LPIPS ↓	Latency (s) ↓
HY-World 1.5	**Diagonal**	**18.4485**	0.7661	**0.1082**	**118.75**
	Full	18.3827	**0.7665**	0.1097	123.10
Causal-Forcing	**Diagonal**	**16.2828**	**0.7464**	0.1554	**101.21**
	Full	16.2759	0.7449	**0.1547**	104.07

C.5 The Effect of the Number of Candidate Centroids in QSAC

QSAC first selects the top-M candidate centroids using the Query-aware distance and then determines the final centroid according to the quantization-aware score. We study the effect of M on HY-World 1.5 and Causal-Forcing in Table 11. Increasing M beyond 4 brings only marginal changes in frame-level quality. In particular, evaluating all 256 centroids does not improve PSNR, SSIM, or LPIPS over $M = 4$ on either model, while significant increasing the inference latency. These results indicate that a small candidate set is sufficient to retain high-quality centroids for the subsequent quantization-aware selection, so we use $M = 4$ throughout our experiments as a practical trade-off between visual quality and efficiency. Note that the latency results in Tables 10 and 11 are mismatched because they are measured in separate cold-start runs with different GPU occupancy. Please compare them in each ablation.

D Efficient Algorithm-System Codesign

D.1 Implementation Details

Figure 9 illustrates how our **Quant*WM*** integrates cache compression, packed storage and on-demand reconstruction into streaming inference.

Cache lifecycle At each native cache-chunk boundary, **Quant*WM*** replaces the completed BF16 chunk with its packed representation and releases the original storage. The active chunk remains in BF16 until completion, while historical chunks remain compressed between attention calls. To support online QSAC and PSAC, we incrementally maintain per-head Query statistics and compute the principal basis at chunk boundaries without retaining historical Query tensors.

Packed storage We physically pack four INT2 residual codes into each byte and store centroid assignments as `uint8` indices. The zero-points are also bit-packed as INT2, while group scales use INT8 codes with a shared FP16 secondary scale. Each INT8 PSAC coefficient row has an FP16 scale. We retain a quantized correction only when it reduces the approximated Query-weighted

Table 11: Effect of the number of candidate centroids M in QSAC. $M = 4$ achieves a favorable trade-off between visual quality and inference cost, while exhaustive evaluation over all 256 centroids provides no additional quality benefit.

Model	M	PSNR ↑	SSIM ↑	LPIPS ↓	Latency (s) ↓
HY-World 1.5	1	18.2442	0.7612	0.1132	116.06
	2	18.3988	0.7639	0.1100	116.22
	4	18.4485	0.7661	**0.1082**	118.75
	8	18.4393	0.7658	0.1089	119.51
	16	**18.4880**	**0.7665**	0.1086	120.65
	32	18.4701	0.7664	0.1084	118.97
	256	18.4450	0.7661	0.1094	240.66
Causal-Forcing	1	16.1767	0.7434	0.1567	135.07
	2	16.2433	0.7478	0.1543	135.84
	4	16.2828	0.7464	0.1554	135.08
	8	16.2805	0.7479	0.1535	144.90
	16	**16.2834**	**0.7487**	**0.1529**	148.43
	32	16.2161	0.7470	0.1554	159.97
	256	16.2067	0.7463	0.1559	1386.38

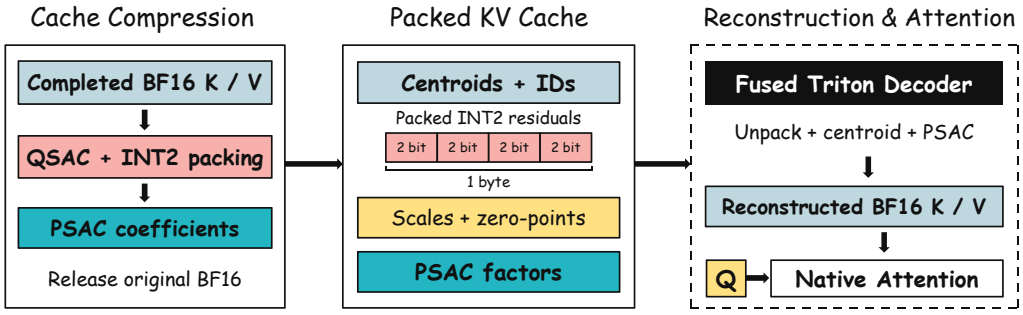


Figure 9: Overview of **Quant*WM***'s inference system. Completed KV cache chunks are stored in a packed low-bit representation. A fused decoder reconstructs BF16 keys and values on demand for native attention.

error; otherwise, its coefficient row is set to zero. For each cache chunk, we select the smaller of dense and sparse coefficient layouts, including the storage cost of sparse indices.

Fused reconstruction During each KV-cache access for attention computation, a Triton kernel fuses residual unpacking, quantization-parameter reconstruction, centroid gathering and PSAC accumulation. Values follow the same reconstruction path without PSAC. The kernel writes only the reconstructed BF16 K/V, which avoids separate dense intermediates for centroid addition and low-rank compensation. The model's native FlashAttention or SDPA implementation consumes these temporary tensors, which are not retained in the persistent cache. We preserve each model's native cache layout, attention mask and RoPE convention, where pre-RoPE Keys receive positional encoding after reconstruction, while post-RoPE Keys are reconstructed directly.

D.2 The Impact of KV Cache Quantization on Latency is Model-dependent

Although KV cache quantization is mainly designed to reduce memory consumption, its impact on end-to-end latency can vary across different system configurations. Online quantization and cache reconstruction introduce additional computation, while the compressed representation reduces the amount of KV data accessed and transferred during generation. For models that are more constrained by KV-cache memory traffic or cache offloading, the reduction in data movement can outweigh the additional quantization overhead and lead to lower latency. LongCat-Video in Table 7 is such a case, where its native inference pipeline offloads a large KV cache, and QuantWM reduces the amount of

cache data repeatedly transferred between host and GPU, which brings lower end-to-end latency. In contrast, when the KV cache is primarily GPU-resident, the reduction in data movement is smaller and the additional quantization operations may introduce a modest latency overhead. Therefore, the latency impact of KV cache quantization depends not only on the compression ratio, but also on how cached representations are accessed and moved during inference.