TRAJQUANT: TRAJECTORY-AWARE QUANTIZATION OF INTERACTIVE VIDEO WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Interactive video world models generate future visual states conditioned on actions and reuse those states as history for subsequent interaction. Their repeated denoising and long autoregressive rollouts make low-bit inference attractive, but conventional local reconstruction does not explicitly account for how quantization errors propagate along action-conditioned trajectories. We study this mismatch as a **control-to-trajectory error** problem: a local perturbation introduced during one chunk alters the inputs encountered by later chunks, while layer-wise tensor reconstruction may fail to capture errors in the complete state transition. We propose **TrajQuant**, a trajectory-aware post-training quantization framework for action-conditioned autoregressive video world models. Its **MLGR** module applies structured rotations whose angles are refined against complete quantized Transformer-block outputs, improving reconstruction at the transition level. Its **FGCC** module (Fisher-Guided Chunk Calibration) estimates a squared-gradient sensitivity proxy for each chunk–layer pair and uses it to weight paired full-precision and upstream-quantized reconstruction, accounting for errors propagated from preceding chunks. MLGR is fixed before FGCC performs final weight rounding in the transformed coordinates. We evaluate Matrix-Game-2 and ABot-World-0 at W6A6, W4A8, and W4A6 with WBench, LPIPS-FP16, and FVD-FP16. On Matrix-Game-2 at W4A6, TrajQuant achieves LPIPS-FP16/FVD-FP16 of 0.1294/32.71 versus SVDQuant's 0.1709/45.26. On ABot-World-0 at W4A6, it also achieves the lowest LPIPS-FP16 and FVD-FP16 among the listed quantized methods, while the leaders at other precisions differ by metric.

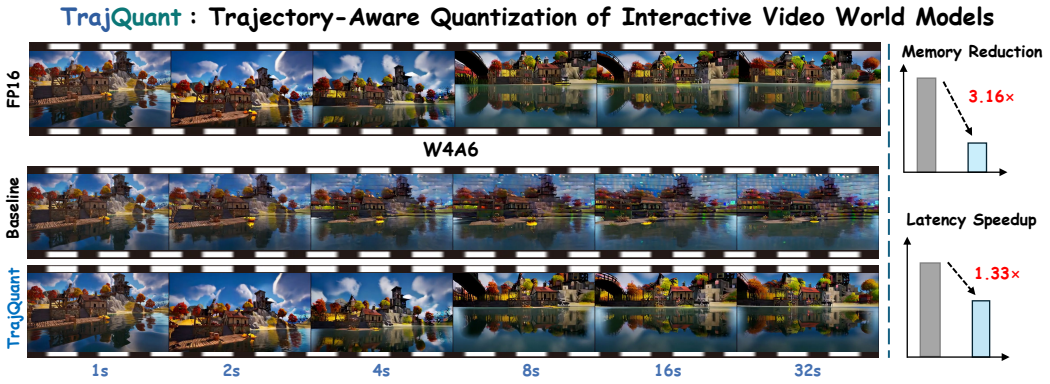

Figure 1: Qualitative comparison of FP16, the baseline quantized model, and TrajQuant across the generated video sequence. The right-side summaries report the measured memory reduction and end-to-end latency speedup.

1 INTRODUCTION

Interactive video world models turn video generation into an action-conditioned simulation: a user or agent supplies controls, the model generates the next visual state, and that state becomes con-

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.

text for subsequent interaction. Genie (Bruce et al., 2024) and Matrix-Game 2.0 (He et al., 2026) illustrate this transition from passive synthesis to controllable environments, while autoregressive video diffusion provides a causal route to streaming generation (Huang et al., 2025). Diffusion-transformer backbones (Peebles & Xie, 2023) require repeated denoising, and long rollouts add to the inference cost, motivating post-training quantization (PTQ).

Existing PTQ methods provide tools for reconstructing weights and reshaping activation distributions. GPTQ (Frantar et al., 2023) uses approximate second-order information for layer-wise weight reconstruction, while SmoothQuant (Xiao et al., 2023), QuaRot (Ashkboos et al., 2024), and ParoQuant (Liang et al., 2026) transform activations and weights before quantization. Diffusion-oriented methods account for timestep variation, activation outliers, conditioning, and video features (Li et al., 2023; Zhao et al., 2025; Ryu et al., 2025; Li et al., 2025; 2026b). Autoregressive visual quantizers also address sample variation and sequential error propagation (Liu et al., 2026; Tang et al., 2026). In an interactive world model, these numerical goals must be considered alongside the action-conditioned transition: quantization should preserve how different controls change the generated future.

The first challenge is choosing a reconstruction target that reflects the model transition. A single-layer output error does not capture interactions among attention, conditioning, feed-forward, and residual paths within a Transformer block. A channel transform that looks favorable for an individual projection may therefore still alter the output of the complete block. We optimize the transform against that block output rather than selecting it from a local activation statistic alone.

The second challenge is trajectory-dependent sensitivity. After generation, chunk c is written into the history used for chunk $c + 1$. Under matched actions and sampling noise, a local approximation separates the newly introduced quantization error from the error carried by the previous history. Some perturbations may be attenuated rather than amplified, but the feedback path makes it important to distinguish the chunk and layer at which an error is introduced. Calibration must also account for the inputs a layer receives after upstream layers have been quantized.

TrajQuant addresses these two levels of error in sequence. MLGR learns multi-stage pairwise Givens rotations by reconstructing complete quantized Transformer blocks. With those transforms fixed, FGCC estimates a squared-gradient sensitivity proxy for each chunk–layer pair on teacher-forced full-precision trajectories. The resulting weights emphasize the corresponding matched full-precision and upstream-quantized inputs during sequential reconstruction. Figure 2 summarizes the two calibration stages. TrajQuant has the lowest FP16-referenced LPIPS and FVD among the listed methods on Matrix-Game-2 at all three precisions and on ABot-World-0 at W4A6; the ABot-World-0 rankings at W4A8 and W6A6 differ by metric.

Our contributions are:

- We formulate block-level reconstruction and chunk–layer sensitivity as complementary requirements for quantizing autoregressive video world models.
- We introduce MLGR, which learns pairwise rotations against complete quantized-block outputs, and FGCC, which weights paired reconstruction by chunk–layer sensitivity in the fixed transformed coordinates.
- We evaluate TrajQuant on Matrix-Game-2 and ABot-World-0 at W6A6, W4A8, and W4A6 using WBench, LPIPS-FP16, and FVD-FP16, with component comparisons at W4A6.

2 Related Work

Interactive Video World Models. Interactive video world models generate observations from actions and visual history. Genie (Bruce et al., 2024) learns a controllable latent action space, while Matrix-Game 2.0 (He et al., 2026) accepts explicit mouse and keyboard inputs for streaming interaction. Because generated observations become subsequent inputs, autoregressive models face a mismatch between clean training histories and generated inference histories; Self Forcing (Huang et al., 2025) addresses that mismatch during training. Quantization introduces a separate deployment-time perturbation into the same feedback path. Our setting therefore couples action-conditioned prediction with the propagation of quantization error through generated history.

Post-Training Quantization of Transformers. Transformer PTQ combines weight reconstruction with equivalent transformations of activations and weights. AdaRound (Nagel et al., 2020) optimizes data-dependent rounding, while BRECQ (Li et al., 2021) extends reconstruction from individual layers to blocks. GPTQ (OPTQ in its conference version) (Frantar et al., 2023) uses approximate second-order information for weight quantization. SmoothQuant (Xiao et al., 2023) balances activation and weight ranges through channel scaling; QuaRot (Ashkboos et al., 2024) uses function-preserving rotations; SpinQuant (Liu et al., 2025) learns rotations for quantized LLMs; and ParoQuant (Liang et al., 2026) parameterizes pairwise Givens rotations. MLGR refines its angles against complete quantized-block outputs in the world-model setting.

Compensation-based quantization accounts for the input mismatch created by preceding quantized layers. ResComp (Li et al., 2026a) decomposes the resulting residual relative to the original weight target. FGCC retains paired full-precision and upstream-quantized inputs while weighting their reconstruction by chunk–layer squared-gradient scores. Those scores come from teacher-forced reference trajectories and serve as a calibration proxy rather than a gradient through future free-running chunks.

Quantization of Diffusion and Autoregressive Visual Models. Diffusion quantizers account for sampling dynamics and nonuniform feature distributions. PTQ4DM (Shang et al., 2023) studies quantized operations and calibration for changing diffusion timesteps, while Q-Diffusion (Li et al., 2023) introduces timestep-aware calibration. For diffusion transformers, Q-DiT (Chen et al., 2025) adapts quantization granularity and activation scales, and ViDiT-Q (Zhao et al., 2025) considers variation across tokens, channels, timesteps, and guidance. DGQ (Ryu et al., 2025) adapts quantization to activation and cross-attention distributions. SVDQuant (Li et al., 2025) absorbs outliers into a high-precision low-rank branch, while DVD-Quant (Li et al., 2026b) combines data-free refinement with rotated scaling and adaptive bit switching. These approaches motivate distribution-aware calibration, whereas TrajQuant additionally reconstructs complete action-conditioned blocks and weights calibration across autoregressive chunks.

Autoregressive visual quantization also models variation across generated samples and time. PTQ4ARVG (Liu et al., 2026) addresses channel-, token-, and sample-wise effects; Q-ARVD (Tang et al., 2026) uses temporal weighting for error propagation in autoregressive video diffusion. FGCC differs in using a separate sensitivity profile for each layer and a matched reconstruction target that includes upstream quantization. Together with MLGR's block-level angle refinement, this makes the calibration objective sensitive to both local block outputs and the position of a chunk in an interaction trajectory.

3 METHOD

We study post-training quantization (PTQ) for action-conditioned autoregressive video world models. The goal is to reduce model memory and inference cost while preserving both the quality of each generated chunk and the behavior of a free-running interaction. This setting differs from one-shot video generation: the output of chunk c is inserted into the history used to predict chunk $c+1$, so a small local perturbation can change the input distribution of later predictions.

We first describe how a local quantization error enters subsequent chunks, then develop the two calibration stages shown in Fig. 2. MLGR learns a multi-stage channel transformation against a complete quantized block, and FGCC uses chunk–layer sensitivity to weight reconstruction after that transformation is fixed. The activation profiles and sensitivity map in the overview illustrate the operations; their values are not measurements.

3.1 TRAJECTORY ERROR AND DESIGN REQUIREMENTS

Let h_c^{fp} and h_c^{q} denote the history and persistent cache before chunk c in the full-precision and quantized runs. We use the same action a_c and sampling randomness ξ_c in both runs, so that the effect of quantization can be separated from a change in the control sequence or the random draw. The two runs evolve according to

$$h_{c+1}^{\mathrm{fp}} = \mathcal{T}_\theta(h_c^{\mathrm{fp}}, a_c; \xi_c), \qquad h_{c+1}^{\mathrm{q}} = \mathcal{T}_{\widehat{\theta}}(h_c^{\mathrm{q}}, a_c; \xi_c). \tag{1}$$

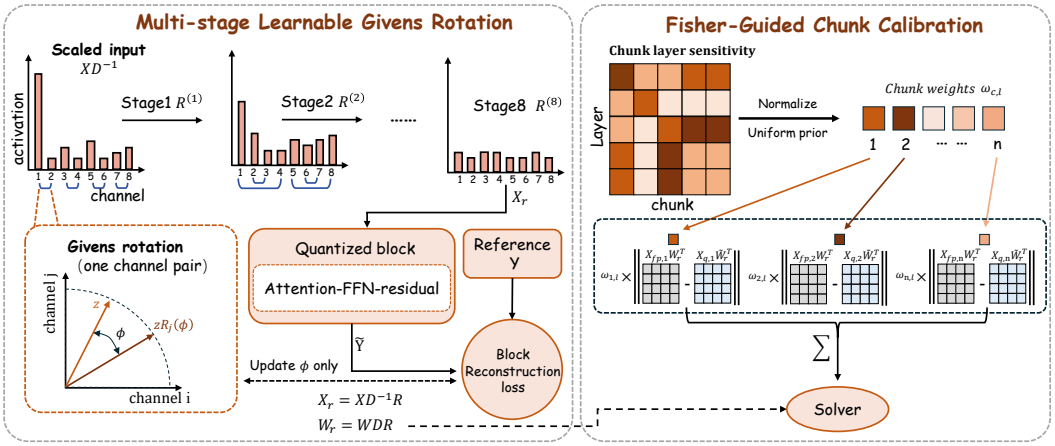

Figure 2: Overview of TrajQuant. MLGR applies multi-stage pairwise Givens rotations and refines their angles using quantized Transformer-block reconstruction. FGCC uses chunk–layer sensitivity to weight paired calibration errors before weight quantization. Activation profiles and sensitivity patterns are schematic.

The quantity of interest is the difference between the two histories, $\delta_c^{\text{hist}} = h_c^{\text{q}} - h_c^{\text{fp}}$. Let ε_c denote the error introduced by applying the quantized model to the full-precision history at chunk c, and let J_c denote the local sensitivity of the next history to a perturbation in the current history. Around the full-precision trajectory, the one-step error therefore satisfies

$$\delta_{c+1}^{\text{hist}} \simeq \varepsilon_c + J_c \delta_c^{\text{hist}}. \tag{2}$$

Here ε_c captures the newly introduced error, while $J_c \delta_c^{\text{hist}}$ captures the discrepancy already present in the history and carried into the next transition. Equation (2) is a local diagnostic, not a claim that errors must grow: J_c can attenuate some directions as well as amplify others. It gives two concrete requirements for quantization. The calibration signal must distinguish which chunk–layer pairs matter, and the reconstruction target must approximate the complete transition that produces the next state rather than an isolated tensor error.

MLGR addresses the block-level requirement by learning structured orthogonal transforms against complete quantized Transformer-block outputs. FGCC addresses trajectory-dependent sensitivity by reweighting paired full-precision and upstream-quantized reconstruction for each chunk–layer pair. The transforms learned by MLGR are fixed before FGCC performs final sequential weight rounding, so reconstruction takes place in the coordinates used at deployment.

For a linear layer, we write $Y = XW^{\mathsf{T}} + b$, where $X \in \mathbb{R}^{N \times d}$ contains token rows and $W \in \mathbb{R}^{m \times d}$. The weight and activation precisions, group size, and activation granularity are configuration choices. We denote the quantization operator by $Q(\cdot)$ and the feasible quantized weights by $\mathcal{Q}$; neither notation fixes a precision.

3.2 MLGR: Multi-stage Learnable Givens Rotation

Equivalent transformations. Before rounding, we apply channel scaling followed by a structured orthogonal transformation. Following SmoothQuant (Xiao et al., 2023), we balance each input channel i using its activation magnitude and the corresponding weight-column magnitude:

$$s_i = \frac{\left(\max_n |X_{ni}|\right)^{\alpha}}{\left(\max_o |W_{oi}|\right)^{1-\alpha}}. \tag{3}$$

Here n indexes calibration tokens and o indexes output channels, consistent with $Y = XW^{\mathsf{T}} + b$. In the implementation, we clip the resulting scales to $[s_{\min}, s_{\max}]$ before setting $D = \text{diag}(s)$. For an orthogonal rotation R, the equivalent transformation is

$$XW^{\mathsf{T}} = (XD^{-1}R)(WDR)^{\mathsf{T}} = X_r W_r^{\mathsf{T}}, \tag{4}$$

where $X_r = XD^{-1}R$ and $W_r = WDR$ are the transformed input and weight. The bias is unchanged. We follow the sparse pairwise Givens parameterization of ParoQuant (Liang et al., 2026), with fixed channel groups and eight rotation stages in the reported configuration. MLGR refines the angles using the complete quantized Transformer-block output as its calibration target.

Figure 3 reports measured activation magnitudes in two selected action-model projections before and after the complete MLGR transform. Their annotated maxima change from 101 to 2.512 and from 126.5 to 1.252, respectively. The plots use different vertical scales, so the annotations, rather than the apparent heights of the surfaces, support this comparison. Because both fixed scaling and learned rotation are applied, these examples do not isolate the effect of the rotation angles.

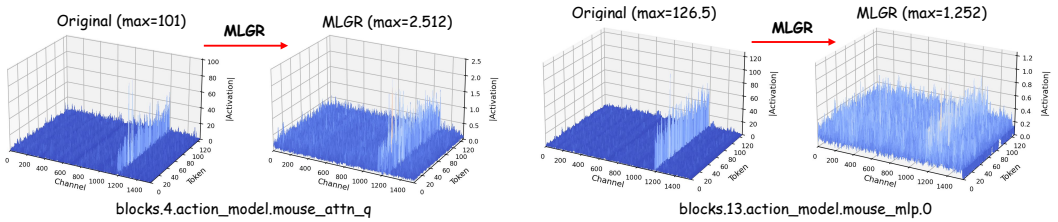

Figure 3: Measured activation magnitudes in two selected action-model projections before and after the complete MLGR transform. The vertical scales differ between panels; each plot reports its maximum in the title.

For a pair of channels, a Givens rotation is

$$G(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}, \qquad R = \prod_{k=1}^{K} R^{(k)}. \tag{5}$$

Each $R^{(k)}$ applies disjoint pairwise rotations within a stage; partners may change between stages. We use $K = 8$ stages in the configuration depicted in Fig. 2. The pair topology remains fixed during calibration. Angles are initialized from activation Gram statistics and then refined using the quantized forward pass. This separates a distribution-based initialization from the block-output objective used to learn the final angles.

Block-level objective. For Transformer block b, let Y_b be the output of the full-precision block and $\widehat{Y}_b(\Phi_b)$ the output of the quantized block with rotation angles Φ_b. The block includes attention, conditioning/action injection, FFN, residual connections, and the weight and activation quantize–dequantize operators. We optimize only Φ_b:

$$\mathcal{L}^{(b)}_{\mathrm{MLGR}}(\Phi_b) = \frac{\|\widehat{Y}_b(\Phi_b) - Y_b\|_F^2}{\mathrm{mean}(Y_b^2) + \epsilon}. \tag{6}$$

The original weights, channel scales, pair topology, quantizer, and model architecture remain fixed. A straight-through estimator is used for rounding. The best parameters on a held-out calibration replay are retained, with the initialization included as a candidate; if refinement does not improve the validation reconstruction error, the initialization is restored.

The objective captures interactions among projections, conditioning, and residual paths that a single-layer reconstruction does not. It remains a teacher-forced block-level objective; its effect on free-running rollout is evaluated separately in Sec. 4.

3.3 FGCC: Fisher-Guided Chunk Calibration

Sensitivity estimation. Let C denote the number of chunks in a calibration trajectory; FGCC does not require a particular value of C. We first generate full-precision trajectories using the calibration action sequences. For a reference chunk c, we keep its history and action fixed, re-noise the corresponding latent $x_{0,c}$ at timestep t, and compute the flow-matching loss

$$\mathcal{L}_{c,t} = \mathrm{MSE}(v_\theta(x_{t,c}, t; h_c, a_c),\ \eta - x_{0,c}), \tag{7}$$

where η is the sampled noise. For every linear layer l, we record the squared task gradient

$$F_{c,l} = \mathbb{E}_{\text{images},\, t,\, \eta}\left[\|\nabla_{W_l}\mathcal{L}_{c,t}\|_F^2\right]. \tag{8}$$

$F_{c,l}$ is an empirical sensitivity proxy: it measures how strongly the current chunk loss changes under a perturbation of layer l. It is not a Hessian, and it does not differentiate through future free-running chunks. We use it as a calibration signal for the current chunk and layer, not as a direct long-horizon objective.

Figure 4 reports measured sensitivity for the nine calibration chunks in the reported run. The heatmap shows that the magnitude and chunk profile differ across layer components. The first chunk has the largest across-layer mean in this measurement, while chunks four and five are also elevated; individual layers need not follow that aggregate ordering. The bar chart averages the plotted sensitivity over layers. The bar chart is a diagnostic summary, not the weight assigned to each chunk: FGCC computes a separate $\omega_{c,l}$ for every layer below. The heatmap's color scale reports values per parameter for visualization, while Eq. (8) defines the layer score used for calibration.

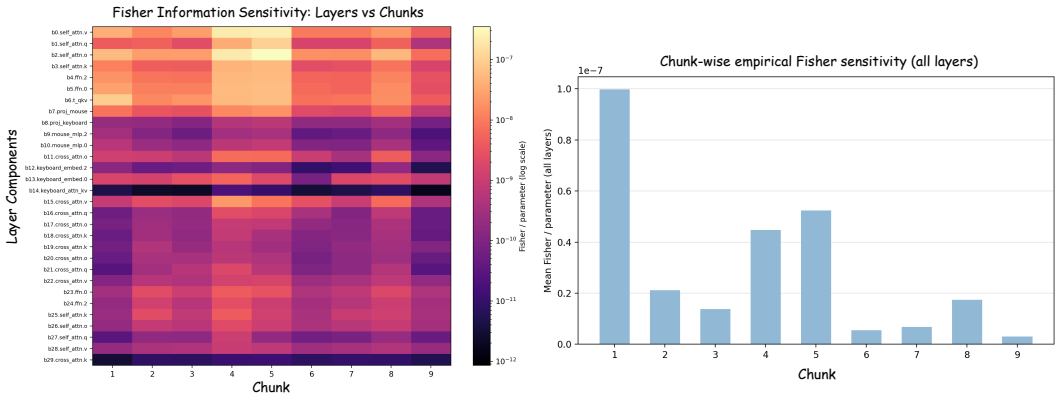

Figure 4: Measured FGCC sensitivity diagnostic for the nine calibration chunks in the reported run. Left: per-parameter squared-gradient values across layer components on a logarithmic color scale. Right: the mean across layers for each chunk, shown as a summary rather than FGCC's layer-specific calibration weights.

For numerical stability, we normalize scores within each layer and combine their softmax distribution with a uniform prior:

$$S_{c,l} = \frac{F_{c,l}}{\max_j F_{j,l} + \epsilon}, \qquad \omega_{c,l} = \frac{1-\lambda}{C} + \lambda \frac{\exp(S_{c,l}/\tau)}{\sum_{j=1}^{C}\exp(S_{j,l}/\tau)}. \tag{9}$$

We use $\lambda = 0.5$ and $\tau = 1$ in the reported configuration. The uniform component prevents a small calibration set from assigning negligible weight to a chunk, while the Fisher component gives additional emphasis to sensitive chunk–layer pairs. Importantly, Eq. (9) does not impose that earlier chunks are always more important; that pattern must be measured from the calibration trajectories.

Weighted reconstruction. After the rotation parameters in Sec. 3.2 are fixed, we quantize the transformed weights layer by layer. The paired full-precision and upstream-quantized inputs $X_{\text{fp},c}$ and $X_{q,c}$ both use the transformed coordinates of Eq. (4): each raw input is multiplied by $D^{-1}R$. For W_r and a candidate quantized weight $\widehat{W}$, we solve

$$\min_{\widehat{W}\in\mathcal{Q}} \sum_{c=1}^{C} \omega_{c,l} \left\| X_{q,c}\widehat{W}^{\mathsf{T}} - X_{\text{fp},c}W_r^{\mathsf{T}} \right\|_F^2. \tag{10}$$

The paired inputs retain the same histories, actions, masks, timesteps, and noise on the two branches. They therefore expose the effect of upstream quantization while avoiding an uncontrolled comparison between two divergent free-running videos.

For sequential rounding, the weighted reconstruction objective is summarized by an input Gram matrix and a cross-input matrix:

$$H_l = \sum_c \omega_{c,l} X_{q,c}^\mathsf{T} X_{q,c}, \qquad G_l = \sum_c \omega_{c,l} X_{q,c}^\mathsf{T} X_{\mathrm{fp},c}. \tag{11}$$

We use a GPTQ-style sequential rounding solver (Frantar et al., 2023) with damping and act-order. The solver operates on W_r and uses the weighted statistics in Eq. (11); it does not quantize an already rounded checkpoint a second time. Following the full-precision target used in ResComp (Li et al., 2026a), let $W^{(k)}$ denote the working weight after rounding step k and let $E^{(k)}$ be its output error relative to the original transformed weight W_r. Then

$$\begin{aligned} E^{(k)} &= X_{\mathrm{fp}} W_r^\mathsf{T} - X_q (W^{(k)})^\mathsf{T} \\ &= (X_{\mathrm{fp}} - X_q)(W^{(k)})^\mathsf{T} + X_{\mathrm{fp}} (W_r - W^{(k)})^\mathsf{T}. \end{aligned} \tag{12}$$

The first term captures the input mismatch caused by upstream quantization; the second captures the difference between the current working weight and the original transformed weight. The chunk weights enter the reconstruction statistics used by the compensation-based solver.

4 Experiments

We evaluate TrajQuant on Matrix-Game-2 and ABot-World-0 at W6A6, W4A8, and W4A6. For each model and precision, we report WBench results and the deviation from the corresponding FP16 generation measured by LPIPS-FP16 and FVD-FP16. We then compare the two components of TrajQuant at W4A6 and report deployment efficiency at W4A8.

4.1 Experimental Setup

Models and rollout protocol. We evaluate Matrix-Game-2 and ABot-World-0. Each model is calibrated using 12 images with seed 42, and FGCC uses nine chunks per calibration trajectory. Matrix-Game-2 uses three sampling steps and ABot-World-0 uses four, following their respective model settings. In evaluation, the quantized model and its FP16 reference receive the same initial state, action sequence, and sampling noise. Each run appends its own generated state to its history, matching the autoregressive inference setting. Preprocessing, chunk size, sampler, and history/cache update order remain fixed within each model. We report the two models separately rather than averaging their scores.

Quantization and comparisons. The three weight/activation precision settings are W6A6, W4A8, and W4A6; the component ablation uses W4A6. The compared methods are symmetric MinMax, SmoothQuant (Xiao et al., 2023), QuaRot (Ashkboos et al., 2024), GPTQ (Frantar et al., 2023), SVDQuant (Li et al., 2025), and ViDiT-Q (Zhao et al., 2025). For TrajQuant, we use $\lambda = 0.5$ and $\tau = 1$ in FGCC. MLGR uses eight stages of pairwise rotations as described in Sec. 3.2. Both calibration stages run offline.

Metrics. WBench (Ying et al., 2026) assesses interactive-video behavior. We report its navigation, spatial consistency, aesthetic quality, background consistency, imaging quality, motion smoothness, dynamic degree, and temporal flickering components separately. LPIPS-FP16 (Zhang et al., 2018) compares quantized outputs with matched FP16 outputs, while FVD-FP16 (Unterthiner et al., 2019) compares their video distributions using the model-specific FP16 generations as reference. Both FP16-referenced distances are lower-is-better. Comparisons within each model use the same evaluation trajectories and FP16 reference.

4.2 Main Results

Matrix-Game-2 results across precisions. Table 1 reports the selected Matrix-Game-2 metrics at W6A6, W4A8, and W4A6. The columns place FP16-referenced distances first, followed by navigation, spatial, aesthetic, background, image, motion, dynamic, and flickering measures. TrajQuant obtains the lowest LPIPS-FP16 and FVD-FP16 at all three precisions. At W4A6, its

Table 1: Matrix-Game-2 results at three weight/activation precisions. Columns follow LPIPS-FP16 and FVD-FP16 (lower is better), then NavScore, spatial consistency, aesthetic quality, background consistency, imaging quality, motion smoothness, dynamic degree, and temporal flickering (higher is better). Gray rows denote TrajQuant. Bold values are best among quantized methods within each precision; ties are all highlighted.

Method	W/A	LPIPS↓	FVD↓	NavScore	Spatial	Aesthetic	Background	Imaging	Motion	Dynamic	Flicker
Matrix-Game-2	16/16	0.0000	0.00	0.8035	0.8831	0.5121	0.9219	0.6583	0.9771	0.9747	0.9290
MinMax	6/6	0.3119	164.04	0.7041	0.8200	0.4940	0.9154	0.6006	0.9049	0.7152	0.8929
SmoothQuant	6/6	0.2362	64.43	0.7699	0.8746	0.5013	0.9146	0.6530	0.9737	0.9684	0.9253
QuaRot	6/6	0.1935	35.77	0.7966	0.8776	0.5090	0.9161	0.6522	0.9752	0.9647	0.9264
GPTQ	6/6	0.3091	164.67	0.7091	0.8201	0.5040	0.9159	0.6311	0.9148	0.7152	0.9027
SVDQuant	6/6	0.1336	33.71	0.8000	0.8771	0.5027	0.9124	0.6587	0.9764	0.9531	0.9284
ViDiT-Q	6/6	0.1200	23.44	0.8010	0.8766	0.5093	0.9106	0.6573	0.9769	0.9647	0.9231
TrajQuant	6/6	**0.0936**	**20.79**	**0.8029**	**0.8833**	**0.5121**	**0.9217**	**0.6588**	**0.9771**	**0.9747**	**0.9290**
MinMax	4/8	0.1828	50.09	0.7933	0.8094	0.4114	0.8211	0.5508	0.8556	0.9684	0.8316
SmoothQuant	4/8	0.1758	45.55	0.7961	0.8529	0.4811	0.9021	0.6073	0.9062	0.9620	0.8802
QuaRot	4/8	0.1791	51.32	0.7964	0.8655	0.5033	0.9142	0.6240	0.9571	0.9684	0.8918
GPTQ	4/8	0.1542	37.80	0.7953	0.8327	0.4517	0.8722	0.5858	0.8767	0.9684	0.8401
SVDQuant	4/8	0.1559	42.83	0.7983	0.8623	0.5027	0.9220	0.6390	0.9764	0.9746	0.9196
ViDiT-Q	4/8	0.1691	45.40	0.7963	0.8726	0.5017	**0.9331**	0.6368	0.9667	0.9620	0.9011
TrajQuant	4/8	**0.1185**	**27.53**	**0.8010**	**0.8846**	**0.5111**	0.9229	**0.6583**	**0.9773**	**0.9747**	**0.9297**
MinMax	4/6	0.3380	169.77	0.7127	0.7067	0.4828	0.8428	0.5533	0.9038	0.7215	0.8630
SmoothQuant	4/6	0.2583	71.31	0.7986	0.8745	0.5113	0.9154	0.6483	0.9730	0.9620	0.9270
QuaRot	4/6	0.2468	60.22	0.7953	0.8786	0.5099	0.9164	0.6483	0.9751	0.9684	**0.9300**
GPTQ	4/6	0.3224	167.73	0.7108	0.7218	0.4946	0.9051	0.6091	0.9546	0.7152	0.9029
SVDQuant	4/6	0.1709	45.26	0.7947	0.8810	**0.5132**	0.9224	0.6550	0.9762	0.9620	0.9206
ViDiT-Q	4/6	0.1840	47.50	0.7957	0.8793	0.5106	0.9205	0.6547	0.9762	0.9684	0.9203
TrajQuant	4/6	**0.1294**	**32.71**	**0.8011**	**0.8844**	**0.5132**	**0.9226**	**0.6582**	**0.9770**	**0.9720**	0.9295

LPIPS-FP16/FVD-FP16 values are $0.1294/32.71$, compared with $0.1709/45.26$ for SVDQuant, the next-best method on both distances. The gap persists as precision changes: at W4A8, TrajQuant reaches $0.1185/27.53$ against GPTQ's $0.1542/37.80$, while at W6A6 it reaches $0.0936/20.79$ against ViDiT-Q's $0.1200/23.44$. Among the quantized methods, TrajQuant leads all eight reported WBench components at W6A6 and seven of eight at W4A8 and W4A6, including a tie in aesthetic quality at W4A6. The exceptions are background consistency at W4A8, where ViDiT-Q scores 0.9331 versus TrajQuant's 0.9229, and temporal flickering at W4A6, where QuaRot scores 0.9300 versus 0.9295. At W4A6, TrajQuant also retains a NavScore of 0.8011 and spatial consistency of 0.8844, compared with GPTQ's 0.7108 and 0.7218.

ABot-World-0 results and their analysis are reported in Appendix Table 4.

Qualitative comparison. Figure 5 juxtaposes W4A6 frame sequences at corresponding positions for FP16, five baselines, and TrajQuant. In this example, TrajQuant retains the road and tree layout across the displayed sequence. The last displayed ViDiT-Q frame contains a dark region not present in the corresponding FP16 and TrajQuant frames. This example makes the temporal differences visible, while the LPIPS-FP16 and FVD-FP16 comparisons above measure the deviation over the evaluation videos.

4.3 Ablation Study

Table 2 compares FGCC alone, MLGR alone, and their combination on Matrix-Game-2 at W4A6. FGCC alone yields LPIPS-FP16/FVD-FP16 of $0.2886/146.84$, whereas MLGR alone reaches $0.1781/42.82$. Combining them further reduces the distances to $0.1294/32.71$ and improves NavScore from 0.7997 for MLGR alone to 0.8011. Spatial consistency rises from 0.8810 to 0.8844, and aesthetic quality from 0.5032 to 0.5132. The MLGR-only configuration also exceeds FGCC alone in NavScore (0.7997 versus 0.7608) and spatial consistency (0.8810 versus 0.7418). Among the

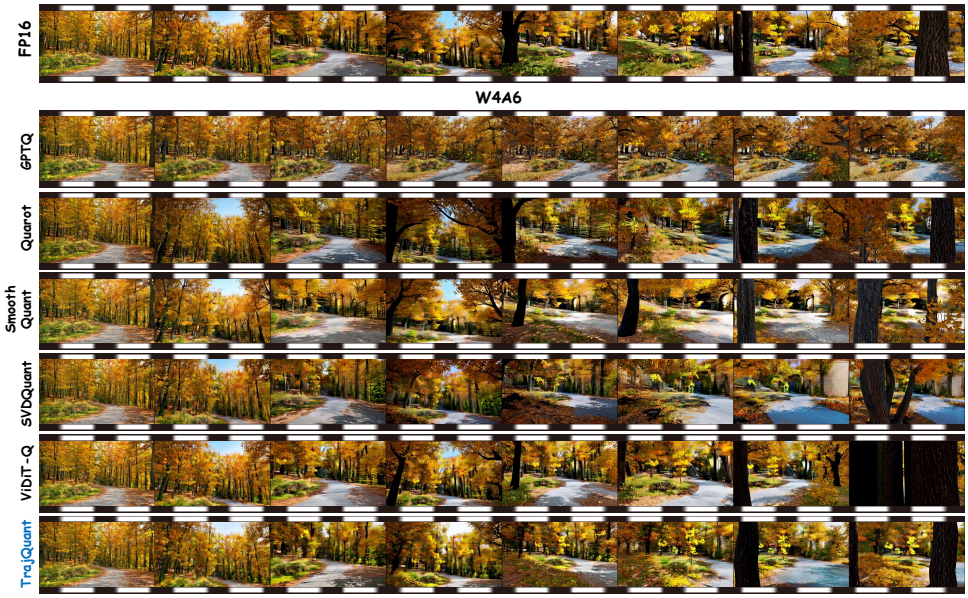

Figure 5: Qualitative W4A6 comparison of a generated video sequence for FP16, five quantization baselines, and TrajQuant. Frames in each column correspond to the same position in the sequence.

Table 2: Component ablation on Matrix-Game-2 under W4A6. LPIPS and FVD are lower-is-better; the remaining metrics are higher-is-better.

Methods		Bit Width (W/A)	LPIPS ↓	FVD ↓	Aesthetic Quality	Nav Score	Spatial Consistency
MLGR	FGCC						
	✓	4/6	0.288559	146.8369	0.5046	0.7608	0.7418
✓		4/6	0.178107	42.8222	0.5032	0.7997	0.8810
✓	✓	4/6	0.129403	32.7134	0.5132	0.8011	0.8844

tested configurations, MLGR alone performs better than FGCC alone on these measures; adding FGCC to MLGR improves all five reported metrics further.

4.4 Deployment Efficiency

Table 3 compares W4A8 TrajQuant with the FP16 Matrix-Game-2 implementation on an H800 GPU. It reduces GPU memory by approximately $3.16\times$ and speeds up end-to-end inference by $1.33\times$. The reported run requires three hours of offline calibration on the same GPU; that cost is excluded from inference latency.

Table 3: Matrix-Game-2 memory and end-to-end latency factors on an H800 GPU (FP16: $1.00\times$); the memory factor is approximate.

Bit-width (W/A)	Memory Opt.	Latency Opt.
16/16	$1.00\times$	$1.00\times$
4/8 (ours)	$3.16\times$	$1.33\times$

5 CONCLUSION

TrajQuant combines block-level learnable rotations (MLGR) and Fisher-guided chunk calibration (FGCC) for interactive video world models. It achieves the lowest FP16-referenced LPIPS and FVD among compared methods at all three precisions on Matrix-Game-2 and at W4A6 on ABot-World-0, with mixed rankings at other ABot precisions. On Matrix-Game-2 at W4A8, it reduces GPU memory by approximately $3.16\times$ and speeds up inference by $1.33\times$.

AI USE STATEMENT

Generative AI tools were used only for literature exploration, reference checking, and language polishing of selected sentences in the final version of the manuscript. The research ideas, overall logic, methodological design and description, and experimental analysis were developed independently by the authors. All technical content, citations, equations, implementations, experimental results, and scientific claims were manually reviewed and verified by the authors. The authors take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Section 4 specifies the models, calibration settings, precision levels, baselines, rollout protocol, and evaluation metrics. The method and its calibration procedure are described in Section 3; additional ABot-World-0 results and metric definitions appear in the appendix.

REFERENCES

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-free 4-bit inference in rotated LLMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/b5b939436789f76f08b9d0da5e81af7c-Abstract-Conference.html.

Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C.Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando De Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Lei Chen, Yuan Meng, Chen Tang, Xinzhu Ma, Jingyan Jiang, Xin Wang, Zhi Wang, and Wenwu Zhu. Q-DiT: Accurate post-training quantization for diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28306–28315, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Chen_Q-DiT_Accurate_Post-Training_Quantization_for_Diffusion_Transformers_CVPR_2025_paper.html.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tcbBPnfwxS.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, Baixin Xu, Hao-Xiang Guo, Kaixiong Gong, Size Wu, Wei Li, Xuchen Song, Yang Liu, Yangguang Li, and Yahui Zhou. Matrix-game 2.0: An open-source real-time and streaming interactive world model, 2026. URL https://arxiv.org/abs/2508.13009.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL

https://proceedings.neurips.cc/paper_files/paper/2025/hash/f4823f831af67a3ef15e41a85434422a-Abstract-Conference.html.

Muyang Li, Yujun Lin, Zhekai Zhang, Tianle Cai, Xiuyu Li, Junxian Guo, Enze Xie, Chenlin Meng, Jun-Yan Zhu, and Song Han. SVDQuant: Absorbing outliers by low-rank component for 4-bit diffusion models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/f34f0630c33be15b8c89426bb8056798-Abstract-Conference.html.

Shuaiting Li, Juncan Deng, Kedong Xu, Rongtao Deng, Hong Gu, Minghan Jiang, Haibin Shen, and Kejie Huang. Rethinking residual errors in compensation-based LLM quantization. In *International Conference on Learning Representations*, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/087b346ca4d6dd6fcd0663ff6c9a6b69-Abstract-Conference.html.

Xiuyu Li, Yijiang Liu, Long Lian, Huanrui Yang, Zhen Dong, Daniel Kang, Shanghang Zhang, and Kurt Keutzer. Q-Diffusion: Quantizing diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17535–17545, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Li_Q-Diffusion_Quantizing_Diffusion_Models_ICCV_2023_paper.html.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the limit of post-training quantization by block reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=POWv6hDd9XH.

Zhiteng Li, Hanxuan Li, Junyi Wu, Kai Liu, Haotong Qin, Linghe Kong, Guihai Chen, Yulun Zhang, and Xiaokang Yang. DVD-Quant: Data-free video diffusion transformers quantization. In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/3bd28dd5cc4e15f9e019da13cc0c4844-Abstract-Conference.html.

Yesheng Liang, Haisheng Chen, Song Han, and Zhijian Liu. ParoQuant: Pairwise rotation quantization for efficient reasoning LLM inference. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/7caf9d251b546bc78078b35b4a6f3b7e-Abstract-Conference.html.

Xuewen Liu, Zhikai Li, Jing Zhang, Mengjuan Chen, Jianquan Li, and Qingyi Gu. PTQ4ARVG: Post-training quantization for autoregressive visual generation models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/81a28be483155f802ddef448d6fc4b57-Abstract-Conference.html.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: LLM quantization with learned rotations. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/e5b1c0d4866f72393c522c8a00eed4eb-Abstract-Conference.html.

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7197–7206. PMLR, 2020. URL https://proceedings.mlr.press/v119/nagel20a.html.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Peebles_Scalable_Diffusion_Models_with_Transformers_ICCV_2023_paper.html.

Hyogon Ryu, NaHyeon Park, and Hyunjung Shim. DGQ: Distribution-aware group quantization for text-to-image diffusion models. In *International Conference on Learning Representations*,

2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a70ee7ea485e4fd36abbfc4adf591c28-Abstract-Conference.html.

Yuzhang Shang, Zhihang Yuan, Bin Xie, Bingzhe Wu, and Yan Yan. Post-training quantization on diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1972–1981, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Shang_Post-Training_Quantization_on_Diffusion_Models_CVPR_2023_paper.html.

Siao Tang, Xinyin Ma, Gongfan Fang, Xingyi Yang, and Xinchao Wang. Q-ARVD: Quantizing autoregressive video diffusion models, 2026. URL https://arxiv.org/abs/2605.21072.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A new metric for video generation. In *ICLR 2019 Workshop on Deep Generative Models for Highly Structured Data*, 2019. URL https://openreview.net/forum?id=rylgEULtdN.

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 *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38087–38099. PMLR, 2023. URL https://proceedings.mlr.press/v202/xiao23c.html.

Kaining Ying, Hengrui Hu, Siyu Ren, Jiamu Li, Fengjiao Chen, Ziwen Wang, Xuezhi Cao, Xunliang Cai, and Henghui Ding. WBench: A comprehensive multi-turn benchmark for interactive video world model evaluation. *arXiv preprint arXiv:2605.25874*, 2026.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html.

Tianchen Zhao, Tongcheng Fang, Haofeng Huang, Rui Wan, Widyadewi Soedarmadji, Enshu Liu, Shiyao Li, Zinan Lin, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. ViDiT-Q: Efficient and accurate quantization of diffusion transformers for image and video generation. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a4a1ee071ce0fe63b83bce507c9dc4d7-Abstract-Conference.html.

A ADDITIONAL ABOT-WORLD-0 RESULTS

Table 4: ABot-World-0 results at three weight/activation precisions, using the same metric order as Table 1. LPIPS-FP16 and FVD-FP16 are lower-is-better; WBench components are higher-is-better. Gray rows denote TrajQuant.

Method	W/A	LPIPS↓	FVD↓	NavScore	Spatial	Aesthetic	Background	Imaging	Motion	Dynamic	Flicker
ABot-World-0	16/16	0.0000	0.00	0.7837	0.8825	0.6206	0.9260	0.6981	0.9693	0.9747	0.9475
MinMax	6/6	0.1895	49.43	0.7843	0.8044	0.4816	0.8248	0.5978	0.9090	0.9747	0.8876
SmoothQuant	6/6	0.1582	36.70	0.7862	0.8667	0.5529	0.9062	0.6785	0.9292	0.9684	0.9075
QuaRot	6/6	0.1348	27.59	0.7816	0.8757	0.6018	0.9160	0.6880	0.9324	0.9647	0.9278
GPTQ	6/6	0.1932	55.30	0.7882	0.8348	0.5514	0.8638	0.6280	0.9194	0.9684	0.8980
SVDQuant	6/6	0.1456	30.36	0.7882	0.8729	0.5723	0.9161	0.6875	0.9493	0.9781	0.9321
ViDiT-Q	6/6	0.3006	25.57	0.7758	0.8782	0.5926	0.9174	0.6987	0.9595	0.9747	0.9478
TrajQuant	6/6	0.1312	104.17	0.7933	0.8849	0.6229	0.9267	0.6988	0.9697	0.9784	0.9481
MinMax	4/8	0.2309	272.29	0.7718	0.8573	0.5159	0.8185	0.6784	0.9221	0.9557	0.9043
SmoothQuant	4/8	0.2084	180.89	0.7756	0.8659	0.6162	0.9189	0.6858	0.9720	0.9684	0.9339
QuaRot	4/8	0.2272	138.88	0.7790	0.8841	0.6125	0.9285	0.6963	0.9712	0.9557	0.9406
GPTQ	4/8	0.1425	222.80	0.7877	0.8875	0.6215	0.9252	0.6947	0.9699	0.9747	0.9288
SVDQuant	4/8	0.1719	135.31	0.7827	0.8813	0.6202	0.9239	0.6971	0.9702	0.9684	0.9392
ViDiT-Q	4/8	0.1790	192.07	0.7817	0.8821	0.6122	0.9252	0.6985	0.9704	0.9620	0.9252
TrajQuant	4/8	0.1936	127.62	0.7856	0.8894	0.6230	0.9268	0.6980	0.9701	0.9714	0.9485
MinMax	4/6	0.2601	87.80	0.7079	0.8540	0.5140	0.8152	0.6756	0.9518	0.9620	0.9041
SmoothQuant	4/6	0.2670	101.40	0.7708	0.8658	0.6172	0.9207	0.6846	0.9634	0.9241	0.9569
QuaRot	4/6	0.2174	60.21	0.7885	0.8877	0.6227	0.9286	0.6965	0.9612	0.9620	0.9507
GPTQ	4/6	0.2017	54.08	0.7880	0.8806	0.6209	0.9227	0.6937	0.9700	0.9747	0.9495
SVDQuant	4/6	0.2161	61.79	0.7843	0.8829	0.6227	0.9275	0.6957	0.9612	0.9584	0.9506
ViDiT-Q	4/6	0.2098	56.78	0.7873	0.8855	0.6211	0.9255	0.6976	0.9602	0.9384	0.9492
TrajQuant	4/6	0.1501	32.91	0.7970	0.8858	0.6227	0.9292	0.6973	0.9704	0.9684	0.9493

ABot-World-0 comparisons. Table 4 reports ABot-World-0 in the same metric order as Matrix-Game-2. At W4A6, TrajQuant achieves the lowest LPIPS-FP16 (0.1501) and FVD-FP16 (32.91) among the listed quantized methods. At W4A8, TrajQuant has the lowest FVD-FP16 (127.62), while GPTQ has the lowest LPIPS-FP16 (0.1425 versus TrajQuant's 0.1936). At W6A6, TrajQuant has the lowest LPIPS-FP16 (0.1312), but its FVD-FP16 (104.17) exceeds ViDiT-Q's 25.57. The FP16-referenced distances therefore have different rankings across the two models and precision settings. At W4A6, TrajQuant's NavScore is 0.7970, versus 0.7880 for GPTQ, whereas its spatial consistency (0.8858) is below QuaRot's 0.8877. The WBench components thus distinguish task behavior from the FP16-referenced visual distances rather than repeating their ordering.

B Evaluation Metrics

LPIPS-FP16 (Zhang et al., 2018) measures frame-level perceptual distance between a quantized rollout and its matched FP16 reference. **FVD-FP16** (Unterthiner et al., 2019) compares the feature distributions of the generated videos with those of the corresponding FP16 videos. Lower values of both metrics indicate closer agreement with FP16; neither is an absolute score against ground-truth videos.

The remaining scores are selected WBench (Ying et al., 2026) components, presented on a 0–1 scale in Tables 1 and 4. Larger values indicate more of the named property, with the interpretation of temporal flickering reversed so that a larger score means *less* visible flicker:

NavScore Summarizes whether the generated camera movement follows the requested navigation and remains consistent across turns; WBench combines navigation accuracy and consistency.

Spatial Consistency Measures whether a scene returns to a similar appearance after a round-trip camera trajectory.

Aesthetic Quality Scores the visual appeal of the generated frames.

Background Consistency Measures the stability of background appearance across the video.

Imaging Quality Scores technical quality within individual frames, including visible degradation and artifacts.

Motion Smoothness Measures the temporal consistency of the estimated motion, rather than its magnitude.

Dynamic Degree Measures how much motion occurs in the video; a higher score indicates more dynamic content, not necessarily smoother motion.

Temporal Flickering Scores the stability of adjacent frames; higher values correspond to less inter-frame flicker.