VAR-Q: Tuning-free KV Cache Quantization for Visual Autoregressive Image and Video Generation

Boxun Xu[1,†*] **Jiaji Lu**[1,†] **Yuxuan Yin**[1] **Zihu Wang**[1]
Ziyue Liu[1] **Yu Wang**[1] **Zirui Liu**[2] **Peng Li**[1]

[1]University of California, Santa Barbara [2]University of Minnesota, Twin Cities

Abstract

Visual autoregressive generation spans two dominant paradigms: next-scale prediction for images and video, and next-frame prediction for video. Both decode block-by-block rather than token-by-token, yielding KV caches whose statistics diverge from those of language models. At scale, KV memory exceeds the weights and limits large-scale deployment caused by a large amount of visual tokens. Our analysis of these caches reveals three findings. First, a small set of *massive-value channels* carries the functional load: ablating these outlier channels collapses generation, whereas ablating a matched set of inlier channels leaves quality intact. Second, the outliers organize along two axes, feature and block, with magnitudes persistent within each cell and shifting across cells. Third, this structure arises from two independent mechanisms: projection bias and RoPE-induced scattering. Guided by these findings, we propose VAR-Q, a tuning-free KV-cache quantization framework that groups along the same two axes. The generation loop provides block boundaries for free, removing the need for outlier detection or calibration. Paired with bit-packing kernels for runtime memory reduction, VAR-Q delivers low-bit quantization across nine models spanning both paradigms, cutting KV memory by up to 78.5% with negligible quality degradation and enabling 6× larger inference batches. Code and demos are available on our Project Page.

1 Introduction

Visual generation has consolidated around two autoregressive paradigms. *Next-scale prediction* decodes a multi-scale token map one resolution at a time, and now spans both image [4, 24] and video [12] synthesis. *Next-frame prediction* decodes a video one frame at a time, conditioned on the past frames [7, 29]. Both decode in semantically coherent *blocks*, a scale or a frame, rather than individual tokens. As representation dimensions increase and generation horizons reach hundreds of frames, the Key-Value (KV) cache accumulated across these blocks rivals or exceeds the model weights themselves, easily saturating single-GPU memory. KV-cache quantization has become the standard remedy for language models [2, 5, 9, 11, 13, 14, 21, 30, 32], where per-channel grouping for keys and per-token grouping for values now compress caches to 2–4 bits with negligible perplexity loss. None of this work targets visual AR. The cache structure of these models is not understood, and recipes calibrated to language models do not transfer: applied directly, they collapse generation.

The KV cache of a visual AR model is not a uniform memory of past tokens. Less than 1% of its outlier channels carry its functional load in a numerical manner. In LongLive [29], zeroing just the top 0.5% of channels by magnitude collapses generation entirely, while zeroing ten times as many

*Correspondence: boxunxu@ucsb.edu
†Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

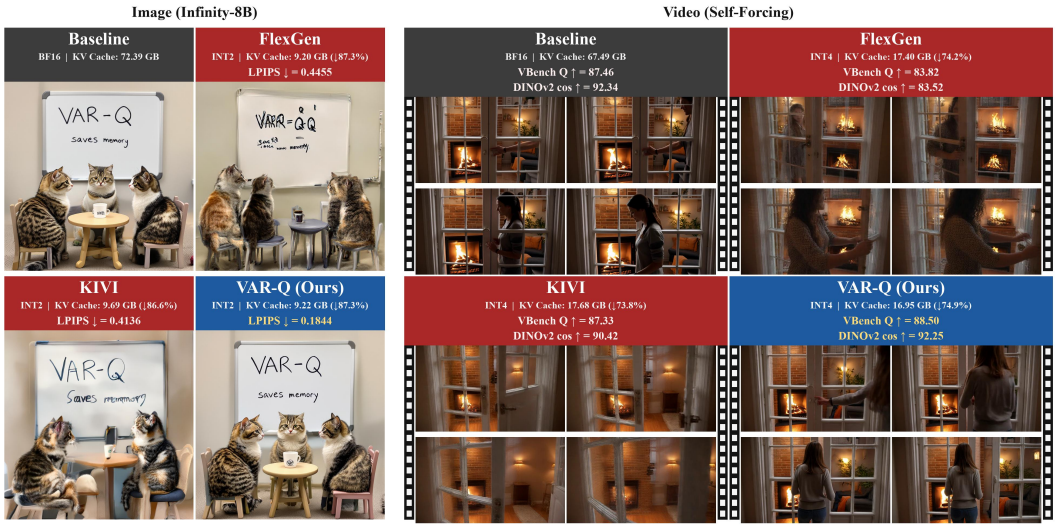

Figure 1: **Generation quality under quantized KV caches.** Top: On Infinity-8B at INT2, `VAR-Q` reproduces the BF16 rendering of text and scene layout that FlexGen and KIVI corrupt. On Self-Forcing at INT4, `VAR-Q` tracks the camera trajectory properly.

inlier channels leaves generation quality largely intact. The same asymmetry holds across Infinity [4], InfinityStar [12], and SelfForcing [7], in both image and video synthesis, and at every layer we measure. Whatever the cache stores for future retrieval lives in a thin, magnitude-distinguished subspace.

This subspace is not a scattered set of outliers but a structured one, organized along two orthogonal axes. A small set of channels carries large magnitudes throughout decoding, producing persistent stripes along the channel axis. Within each block magnitudes are stationary, and across block boundaries they shift sharply, producing bands along the block axis. Both axes have architectural causes. On the channel axis, key outliers arise from a composition of three multiplicative sources: heavy-tailed $\mathbf{W}_K$ row norms seed an outlier baseline, post-projection RMSNorm gain γ_K amplifies it where present, and RoPE pair rotation [19] redistributes energy at the pair level. On the block axis, position-varying RoPE projects channel magnitudes through a sinusoidal modulation, especially at high-frequency channels in the block index, producing the cross-block shifts directly from the rotation schedule rather than from any change in input. Value caches exhibit the same two-axis structure with weaker per-axis attribution, sufficient to justify the same grouping rule without a separate mechanism.

However, existing quantization rules align with at most one axis: KIVI [14] groups keys per-channel and values per-token, leaving the block axis unprotected on both. To this end, we propose **`VAR-Q`**, a tuning-free KV quantization framework that groups the cache jointly along channels and blocks, with one quantization unit per channel and block pair. The block boundaries come for free from the generation loop itself, without detection, calibration, or retraining. We pair the algorithm with bit-packing kernels that turn nominal bit-width reduction into actual memory savings on current GPUs. Across nine visual AR models spanning both paradigms, `VAR-Q` matches the BF16 baseline at INT4 on both next-scale image and next-scale video generation, and remains usable on Infinity-8B down to INT2. On next-frame video, it sustains quality at INT4 and outperforms KV quantization work transferring from LLM domains directly such as KIVI [14] and FlexGen [18]. Complementary to concurrent cache eviction [10, 28] and attention sparsification [26] for visual generation, `VAR-Q` reduces KV memory by up to 78.5% relative to BF16, and inference batch size grows by up to $6\times$ on a single NVIDIA A100 80GB.

2 Background

We unify the two visual autoregressive paradigms under a common block-wise notation. *Next-scale AR* [4, 12, 24] emits one resolution of a multi-scale token map per step; *next-frame AR* [7, 29] emits

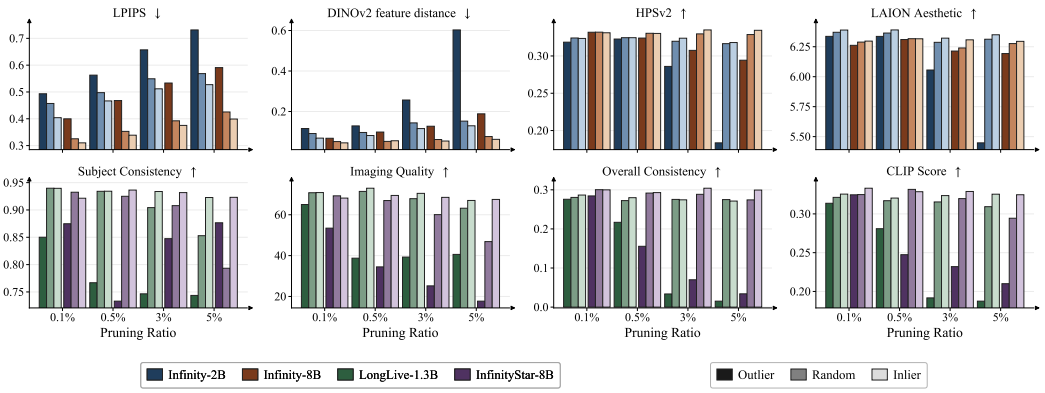

Figure 3: Channel-wise outlier, random, and inlier KV ablation on four visual AR models. Channels are ranked by maximum absolute value over cached tokens.

one or more frames per step, and we treat each frame as a separate block. We refer to the tokens emitted at step i as block i. After B steps, the key cache is

$$\mathbf{K} = \left\{\mathbf{K}^{(i)}\right\}_{i=1}^{B}, \qquad \mathbf{K}^{(i)} \in \mathbb{R}^{n_i \times D}, \tag{1}$$

where n_i is the number of tokens in block i and D is the head-channel dimension; the value cache $\mathbf{V}$ is defined analogously. In next-scale AR, n_i grows geometrically across a fixed number of scales; in next-frame AR, n_i is constant and B grows with the video length. Both caches are produced by linear projections $\mathbf{K}^{(i)} = f_K(\mathbf{x}^{(i)}\mathbf{W}_K)$ and $\mathbf{V}^{(i)} = \mathbf{x}^{(i)}\mathbf{W}_V$, where f_K denotes any model-specific post-projection transforms applied to keys.

3 Methodology

The element-wise magnitudes of the KV cache form a two-axis grid over feature channels and decoding blocks, stable within cells and shifting sharply across block boundaries. A small subset of cells carries values orders of magnitude larger than the rest, and these dominate generation. Group-wise quantization must therefore align with both axes. Section 3.1 establishes that the massive values are critical for generation through controlled ablation; Section 3.2 characterizes the two-axis grid; Section 3.3 attributes both axes to specific architectural sources; and Section 3.4 translates the grid into `VAR-Q`.

3.1 Massive Values in KV Caches

Outliers in the KV cache are functional, not just numerical. Massive activations have been documented in the residual streams [22] and MLP weights [31] of language models, where direct ablations confirm them as functionally indispensable. KV cache outliers have been observed across LLMs [5, 14] and VLMs [20], motivating protect-by-isolation strategies such as per-channel grouping [14], rotation-based redistribution [1], and high-precision retention for dense outliers [5]. This protection establishes utility, not necessity: per-channel grouping and rotation succeed whether outliers carry load-bearing signal, dominate the quantization range of co-grouped channels, or both. Only direct ablation can adjudicate. Visual AR has had neither: no characterization of its KV cache outliers, no ablation of their function. Whether load-bearing outliers exist in this regime, and how they are organized across long visual sequences, remains open.

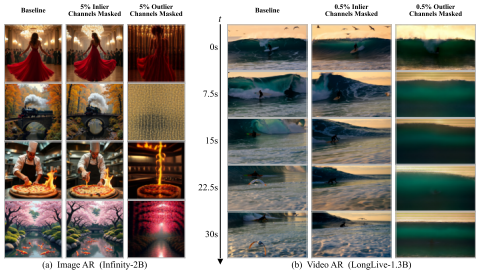

Figure 2: Generation Comparison under channel-wise outlier vs. inlier KV masking.

We test necessity directly through ablation. For each layer, we rank channels by their maximum magnitude across the cached sequence. The top p percent form the *outlier* set. From the remaining

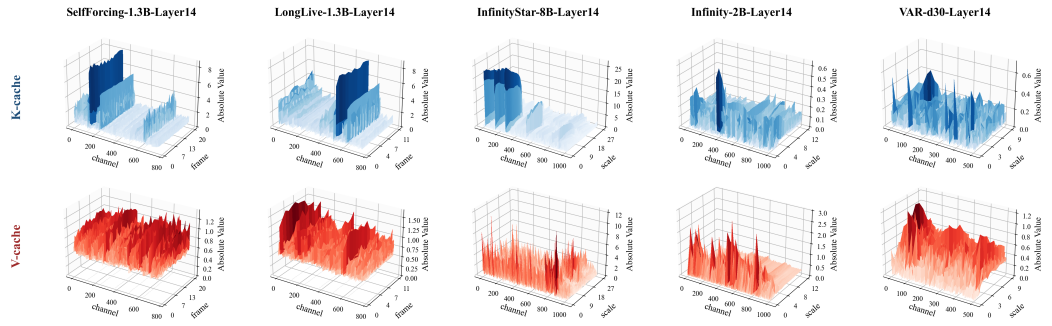

Figure 4: Median absolute values of K (top) and V (bottom) caches across five visual AR models over the (channel, generation index) grid. Channel stripes persist along the generation axis in all models, and their magnitudes shift sharply at block boundaries. Color scales are per-panel.

channels, we draw an equal-sized random sample to form the *inlier* set. Each ablation zeros one set. Figure 2 makes the contrast visible: on Infinity-2B at $p = 5\%$ and LongLive-1.3B at $p = 0.5\%$, outlier ablation collapses generation into uniform texture or solid blocks; inlier ablation leaves outputs intact. Figure 3 extends this to four models and eight metrics. On LongLive-1.3B, masking the top 0.5% drops CLIP from 0.32 to 0.28, while ablating ten times as many inlier channels leaves CLIP at baseline (0.32). We call this small, magnitude-defined subset the *massive-value channels*. [2]

3.2 Massive Values are Structured Along Two Axes in KV Caches

Massive values impose a structural constraint on group-wise quantization: any group that mixes massive and regular cells must stretch its scaling factor to cover the massive cell, starving the regular cells of resolution. Resolving this requires group boundaries that follow the spatial distribution of massive values. In visual AR KV caches, that distribution is two-dimensional: massive values concentrate along a small set of channels and a small set of blocks. Both axes must be covered.

Channel stripes and block bands. The same structure appears in every model in Figure 4, organized along two orthogonal axes. A small subset of channels carries large magnitudes throughout the entire decoding trajectory, producing persistent stripes along the channel axis. Within each block, magnitudes are approximately stationary; across block boundaries, they shift sharply, producing bands along the block axis. The two axes are not redundant; they compose. Massive values live precisely at the intersections of high-magnitude channels and high-magnitude blocks. The block axis is specific to visual AR: language-AR KV caches form a homogeneous token stream with no analogous boundaries [5, 14], and prior LLM quantization methods accordingly exploit only the channel axis.

Table 1: INT4 Quantization Errors (%) on Infinity-2B, where $\epsilon_X = \|X - X'\|_F / \|X\|_F$.

	Reconstruction		Attention	
Rule	ϵ_K	ϵ_V	ϵ_A	ϵ_O
Per-tensor	24.01	43.54	29.20	32.32
Per-block	21.59	32.93	26.36	22.78
Per-channel	12.11	18.07	16.17	13.13
Per-(ch., blk.)	**10.04**	**15.28**	**13.38**	**10.13**

Aligning with both axes minimizes error. Table 1 measures how well four grouping rules contain the structure on Infinity-2B. Per-tensor collapses the entire cache into one group and incurs the largest error on every metric. Per-block aligns with the generation axis but lets each block mix high- and low-magnitude channels. Per-channel aligns with the hidden axis, the choice of KIVI [14] for LLM key caches, but forces a massive cell to share its scaling factor with regular cells of the same channel across all blocks. Only per-(channel, block) aligns with both axes, placing each massive cell in a group that contains only other massive cells from the same channel and the same block. Errors fall monotonically as more axes are covered, and the joint rule wins on every column: ϵ_O from 32.32% to 10.13%, ϵ_K from 24.01% to 10.04%.

[2] A per-tensor decomposition of the ablation reveals that video models are more sensitive to **K** outliers and image models to **V** outliers (Appendix B).

3.3 Why Do Massive Values Emerge in KV Caches of Visual Autoregressive Models?



Figure 5: Outlier persistence across different prompts

While Section 3.2 establishes *where* massive outliers concentrate, this section investigates *why* they emerge. We attribute these structural anomalies along the channel and block axes to specific operations within the K and V projection chains. Formally, we decompose the generation of K_t and V_t as follows:

$$K_t = \mathrm{RoPE}_t\big(\mathrm{RMSNorm}(x_t W_K^\top) \odot \gamma_K\big), \quad V_t = x_t W_V^\top. \tag{2}$$

K outliers are architectural and V outliers are largely input-driven. Whether an outlier is architectural or input-driven can be tested empirically through *cross-prompt persistence*: an architectural outlier should persist across prompts, while an input-driven outlier should drift. We sample N prompts across models and define outliers as the top-p channels by magnitude. **Persistence@N** is the fraction of a single prompt's outlier channels that remain outliers under *all* N prompts (random ≈ 0). At $N{=}10$, $p{=}1\%$, K Persistence@10 reaches $\mathbf{0.53}$–$\mathbf{0.62}$, a hard architectural core shared across inputs; V sits at $\mathbf{0.07}$–$\mathbf{0.14}$, a $\mathbf{3.8}\times$ –$\mathbf{8.5}\times$ gap(Fig. 5). V's near-vanishing persistence indicates input activations dominate V outlier identity.

A Compounding Architectural Effect on K-Cache Channel Outliers. The architectural core of K identified above arises from three multiplicative sources along the projection chain (Eq. (2)): **(i)** heavy-tailed row norms of W_K seed an initial outlier baseline; **(ii)** where RMSNorm-K is present, the channel gain γ_K amplifies this baseline; and **(iii)** RoPE rotates each $(2i, 2i{+}1)$ pair, redistributing energy between the two channels while preserving the pair's ℓ_2 norm.

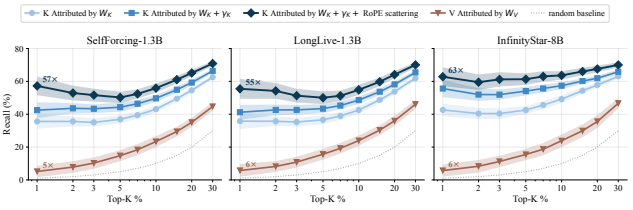

Figure 6: Recall of K outliers as projection-chain components are progressively added; W_V shown as control.

We quantify this compounding by progressively adding components and measuring the recall of final K outliers (Fig. 6). At $p{=}1\%$, W_K alone recovers 35–42% (35–42× random); adding γ_K lifts recall further; the full chain reaches $\mathbf{55}$–$\mathbf{63}\times$ random. In contrast, W_V alone yields 5–6× random recall on every model, an order of magnitude weaker than the K chain.

Block-Axis Dynamics generated by RoPE Periodicity. A class of K block-axis outliers is generated by RoPE alone, independent of input dynamics. RoPE rotates each channel pair $(2i, 2i{+}1)$ at frequency ω_i by an angle $(b\,\omega_i)$ along the pair's RoPE axis, where b is the block index. Within block b, every token shares the same rotation, and per-token magnitudes are

$$|K_{2i}(b,t)| = M_t\,|\cos(b\,\omega_i + \phi_t)|, \qquad |K_{2i+1}(b,t)| = M_t\,|\sin(b\,\omega_i + \phi_t)|, \tag{3}$$

with M_t the pre-RoPE pair amplitude at token t and ϕ_t its initial angle. Let $c_{2i}(b)$ denote the channel's per-token magnitudes within block b, with $\|c_{2i}(b)\|_\infty := \max_t |K_{2i}(b,t)|$. Two consequences follow and are visible in Fig. 7. *(i) Pair envelope is approximately block-invariant.* RoPE preserves the per-token pair ℓ_2 exactly, and in the massive-value regime this carries over to the pair of channel maxes (Appendix A). The two channels move in anti-phase within each block. *(ii) Fast pairs dominate the block axis.* The total angle a pair traverses across the block sequence determines its swing: slow pairs barely rotate and stay flat, fast pairs complete multiple cycles and produce block bands.

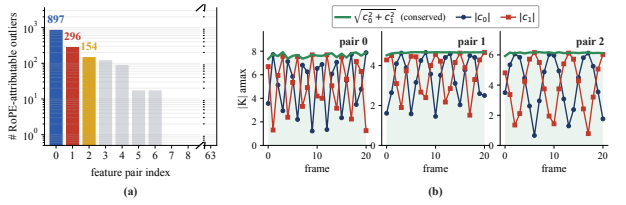

Figure 7: **(a)** On Self-Forcing, block-axis outliers concentrate on the lowest-index (fastest-rotating) pairs. **(b)** Within such a pair, $|c_0|$ and $|c_1|$ swap energy across blocks while pair ℓ_2 stays approximately flat.

3.4 VAR-Q: Algorithm and System Support

Algorithm. Section 3.2 shows that massive values in the KV cache concentrate along two axes: channels and blocks. VAR-Q aligns the quantization grouping with both. As shown in Fig. 8, on top of the per-channel grouping, VAR-Q splits each group at block boundaries. The resulting quantization unit is indexed by (i, j), where i is the block index and j is the channel; each unit carries a single scaling factor under uniform symmetric quantization.

The rule is online without requiring auxiliary detection or calibration and retraining. It is paradigm-agnostic, applying identically whether a block is a scale or a frame. And it is unified across keys and values, since both caches share the same two-axis structure.

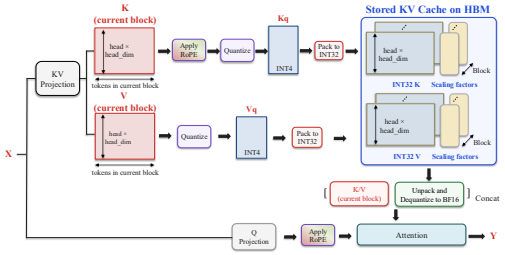

Figure 8: The proposed VAR-Q pipeline.

We further expose a single hyperparameter $r \in (0, 1]$, the sub-block ratio. Let $L_{\max}$ denote the largest block size in the sequence. Each block of size L is partitioned along the token axis into $\lceil L/(rL_{\max}) \rceil$ equal-sized sub-blocks; smaller r yields finer groups. In next-scale AR earlier scales are geometrically smaller than $L_{\max}$ and remain unsplit at typical r, so r effectively controls the granularity of the last scale; in next-frame AR every block has size $L_{\max}$, so r partitions every frame uniformly. We default to $r = 1$ throughout, recovering the per-block rule, and ablate r in Section 4.

System Support. Quantization alone does not reduce the KV-cache footprint, because current CUDA toolchains lack native sub-8-bit storage types. We pair the algorithm with a hardware-friendly Triton implementation that packs sub-byte quantized values into byte-aligned containers on write and unpacks them during attention; details in Appendix F.

4 Experiments

We evaluate VAR-Q on nine models across five architectures, covering both paradigms and modalities: Infinity-2B/8B [4] for next-scale image generation, InfinityStar-8B [12] at 720p and 480p for next-scale video, and Self-Forcing [7] on 5-second video clips and LongLive [29] on 30-second video clips for next-frame video generation. We additionally include VAR-d20/d24/d30 [24] on class-conditional ImageNet-1K. We compare against KIVI [14] and FlexGen [18]. KIVI uses per-channel grouping for K and per-token grouping for V; FlexGen uses 16×16 tile-wise grouping; VAR-Q uses per-channel and per-block grouping with $r = 1$. All methods share identical bit-packing kernels, and each method's scale-factor overhead is included in the reported KV. For class-conditional image generation we report FID, IS, Precision, and Recall, together with PSNR, SSIM, and LPIPS against the BF16 output under the same seeds, isolating quantization fidelity from sampling stochasticity. For text-to-image we report GenEval [3] and DPG [6]; for text-to-video we report VBench [8] and CLIP Score on MovieGen prompts [17], and additionally report VLM-as-judge [33] scores using Claude Opus 4.7 on long-video clips. The tuning-free methods require no calibration data, and INT8 packed kernels are implemented in Triton and evaluated on NVIDIA A100 80 GB.

4.1 Next-Scale Image Generation

Table 2 reports next-scale image generation on VAR (class-conditional) and Infinity (text-to-image).

On VAR-d20/d24/d30, VAR-Q at INT4 matches BF16 in FID across all three depths and leads KIVI and FlexGen on PSNR, SSIM, and LPIPS computed against matched-seed BF16 outputs. FID is noisy at this scale; the consistent PSNR/SSIM/LPIPS advantage shows that VAR-Q tracks the BF16 generation trajectory at the sample level rather than coincidentally landing on a similar distribution. The gap sharpens at INT3: VAR-Q stays within 1.0 FID of BF16 (3.77/3.60/3.05) while KIVI degrades by 1.2 to 2.3 FID and FlexGen collapses (FID 15.3/14.9/9.5).

On Infinity-2B and Infinity-8B, VAR-Q preserves GenEval and DPG Overall within 0.3 points of BF16 at INT4. The 8B variant additionally holds at INT3 (GenEval 83.28, $\Delta = 0.26$) while KIVI

Table 2: Next-scale image generation quality under low-precision KV-cache quantization on VAR-d20/d24/d30 and Infinity-2B/8B. Our INT4 results closely match the BF16 baseline. Bold/underline: best/second-best per precision. KV (GB): realized cache size; subscripts: savings vs. BF16.

(a) VAR-d20/d24/d30 [24] on class-conditional ImageNet 256×256.

Model	#Params	Method	Prec.	Generation Quality				Fidelity to BF16			KV (GB)↓
				FID ↓	IS ↑	Precision ↑	Recall ↑	PSNR ↑	SSIM ↑	LPIPS ↓	
VAR-d20	600M	Baseline	BF16	4.11	357.32	0.8650	0.5160	–	–	–	$16.60_{\text{bs=128}}$
		FlexGen	INT4	4.03	325.39	0.8326	0.5230	11.84	0.264	0.553	**4.29**$_{-74.14\%}$
		KIVI	INT4	3.55	341.52	0.8458	**0.5363**	11.81	0.256	0.552	4.65$_{-71.97\%}$
		VAR-Q	INT4	**3.44**	**342.77**	**0.8502**	0.5281	**13.22**	**0.308**	**0.462**	4.64$_{-72.06\%}$
		FlexGen	INT3	15.33	139.59	0.5944	**0.6308**	10.35	0.212	0.650	**3.46**$_{-79.14\%}$
		KIVI	INT3	6.36	233.84	0.7260	0.5748	9.88	0.187	0.662	3.82$_{-76.97\%}$
		VAR-Q	INT3	**3.77**	**262.24**	**0.7706**	0.5767	**12.24**	**0.266**	**0.538**	3.81$_{-77.06\%}$
VAR-d24	1.0B	Baseline	BF16	2.97	357.24	0.8444	0.5605	–	–	–	$23.91_{\text{bs=128}}$
		FlexGen	INT4	2.74	325.69	0.8175	**0.5816**	11.78	0.256	0.560	**6.18**$_{-74.14\%}$
		KIVI	INT4	2.56	340.55	0.8295	0.5786	11.92	0.268	0.545	6.70$_{-71.97\%}$
		VAR-Q	INT4	**2.52**	**341.60**	**0.8321**	0.5718	**13.03**	**0.296**	**0.479**	6.68$_{-72.06\%}$
		FlexGen	INT3	14.91	144.09	0.6066	**0.6424**	10.63	0.201	0.643	**4.99**$_{-79.14\%}$
		KIVI	INT3	4.29	239.40	0.7271	0.6340	10.88	0.226	0.631	5.51$_{-76.97\%}$
		VAR-Q	INT3	**3.60**	**258.33**	**0.7477**	0.6130	**11.90**	**0.252**	**0.558**	5.48$_{-77.06\%}$
VAR-d30	2.0B	Baseline	BF16	2.07	335.10	0.8260	0.5812	–	–	–	$37.35_{\text{bs=128}}$
		FlexGen	INT4	2.00	307.92	0.8061	**0.6006**	12.47	0.277	0.510	**9.66**$_{-74.14\%}$
		KIVI	INT4	1.99	321.46	**0.8162**	0.5974	12.86	0.290	0.484	10.47$_{-71.97\%}$
		VAR-Q	INT4	**1.97**	**323.67**	0.8156	0.5961	**13.54**	**0.312**	**0.437**	10.44$_{-72.06\%}$
		FlexGen	INT3	9.52	180.57	0.6738	**0.6417**	11.33	0.232	0.597	**7.79**$_{-79.14\%}$
		KIVI	INT3	3.26	264.93	0.7536	0.6189	11.69	0.249	0.573	8.60$_{-76.97\%}$
		VAR-Q	INT3	**3.05**	**268.16**	**0.7612**	0.6307	**12.41**	**0.270**	**0.517**	8.57$_{-77.06\%}$

(b) Infinity-2B/8B [4] on text-to-image generation.

Model	Method	Prec.	GenEval [3] ↑				DPG [6] ↑						KV (GB)↓
			Two-Object	Position	Color	Overall	Entity	Relation	Attribute	Global	Other	Overall	
Infinity-2B	Baseline	BF16	80.61	25.51	88.03	69.61	86.55	89.51	86.62	83.24	90.14	81.97	$41.10_{\text{bs=8}}$
	VAR-Q	INT8	79.34	26.26	87.23	69.06	87.69	89.39	88.70	86.40	85.85	82.21	20.65$_{-49.75\%}$
	FlexGen	INT4	78.57	24.75	54.50	67.75	87.39	**91.56**	85.38	82.24	82.04	81.01	10.60$_{-74.22\%}$
	KIVI	INT4	**80.61**	24.49	**87.23**	67.48	**88.18**	91.13	86.51	83.18	80.81	80.04	10.78$_{-73.76\%}$
	VAR-Q	INT4	79.34	**26.01**	86.97	**69.35**	87.24	89.60	**87.43**	**86.75**	**90.13**	**82.02**	**10.38**$_{-74.75\%}$
	KIVI	INT3	77.30	**26.26**	86.70	67.56	87.12	**92.39**	85.66	82.94	79.31	80.76	8.73$_{-78.76\%}$
	VAR-Q	INT3	**78.32**	24.24	**88.03**	**68.04**	**87.74**	89.92	**87.01**	**85.13**	**87.01**	**81.18**	**8.32**$_{-79.75\%}$
Infinity-8B	Baseline	BF16	95.41	62.12	90.43	83.54	88.27	89.30	91.57	92.83	88.61	85.72	$67.43_{\text{bs=6}}$
	VAR-Q	INT8	96.43	62.88	89.89	84.00	90.57	92.57	89.23	87.12	92.68	85.97	33.88$_{-49.75\%}$
	FlexGen	INT4	95.41	60.89	89.02	82.45	90.75	92.33	89.15	85.04	84.49	84.58	17.38$_{-74.22\%}$
	KIVI	INT4	95.66	**65.40**	91.22	83.26	**91.13**	**94.10**	88.93	85.67	82.04	85.74	17.69$_{-73.76\%}$
	VAR-Q	INT4	**95.92**	63.64	**91.49**	**83.67**	90.70	91.99	**91.07**	**87.47**	**89.23**	**85.90**	**17.03**$_{-74.75\%}$
	KIVI	INT3	**95.66**	**63.38**	**91.49**	82.75	87.74	**94.33**	88.97	84.05	85.71	84.89	14.32$_{-78.76\%}$
	VAR-Q	INT3	95.41	63.13	90.69	**83.28**	**89.65**	88.84	**90.82**	**89.16**	**91.80**	**85.89**	**13.65**$_{-79.75\%}$
	VAR-Q	INT2	85.97	56.57	86.70	74.38	87.80	90.09	89.31	91.22	88.12	83.83	8.60$_{-87.25\%}$

loses 0.79 points, and remains usable at INT2 (GenEval 74.38, DPG 83.83), confirming that larger backbones tolerate more aggressive KV-cache quantization.

4.2 Next-Scale and Next-Frame Video Generation

We evaluate VAR-Q on next-scale video with InfinityStar and on next-frame video with Self-Forcing and LongLive; all three backbones use RoPE.

InfinityStar. At INT4 on both 720p and 480p, VAR-Q matches BF16 in VBench Total Quality (81.66 and 81.32 vs. 82.33 and 81.30) and ranks first on Total at every precision and resolution. The gap widens at INT3 on the higher-resolution 720p setting: KIVI loses 6.87 points to 75.46, while VAR-Q stays within 1.74 points of BF16 at 80.59. KV memory drops by 73 to 79% across precisions.

Table 3: **Next-scale video generation on InfinityStar.** KV reports cache size (GB) with reduction vs. BF16 in subscript. **Bold** and underline indicate best and second-best, respectively.

Model	Method	Prec.	VBench [8] ↑								KV (GB)↓
			Subject Consistency	Background Consistency	Temporal Flickering	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Imaging Quality	Total Quality	
InfinityStar 8B 720p	Baseline	BF16	93.69	94.62	97.71	98.86	67.50	58.49	66.79	82.33	$62.72_{\text{bs=3}}$
	FlexGen	INT4	**93.83**	**94.76**	**97.83**	**98.91**	63.00	55.50	**64.80**	80.11	$16.48_{-73.72\%}$
	KIVI	INT4	92.01	94.61	97.27	98.70	68.00	56.12	63.60	81.24	$16.73_{-73.33\%}$
	VAR-Q	INT4	92.65	94.09	97.37	98.59	**74.50**	**56.87**	64.35	**81.66**	**16.00**$_{-74.49\%}$
	FlexGen	INT3	90.12	92.68	**97.84**	96.91	62.00	**56.01**	60.48	79.95	$13.56_{-78.37\%}$
	KIVI	INT3	90.45	90.42	97.65	96.79	74.00	50.97	**61.92**	75.46	$13.81_{-77.99\%}$
	VAR-Q	INT3	**91.41**	**93.56**	97.25	**98.37**	**75.00**	55.55	61.75	**80.59**	**13.08**$_{-79.15\%}$
InfinityStar 8B 480p	Baseline	BF16	93.01	94.53	97.25	98.68	66.00	58.13	64.01	81.30	$55.87_{\text{bs=6}}$
	FlexGen	INT4	90.59	94.39	95.16	98.64	66.50	56.48	62.14	78.94	$15.03_{-73.10\%}$
	KIVI	INT4	92.44	94.42	96.65	97.79	64.00	56.97	62.92	80.46	$15.25_{-72.71\%}$
	VAR-Q	INT4	**92.75**	**94.51**	**97.21**	**98.67**	**68.50**	**57.84**	**63.62**	**81.32**	**14.60**$_{-73.86\%}$
	FlexGen	INT3	91.89	93.91	97.24	98.62	64.00	57.54	**63.69**	80.64	$12.45_{-77.71\%}$
	KIVI	INT3	**93.01**	**94.46**	**97.33**	**98.71**	64.00	**57.79**	63.56	78.05	$12.67_{-77.33\%}$
	VAR-Q	INT3	92.86	94.30	97.26	98.64	**65.50**	57.52	63.37	**80.98**	**12.03**$_{-78.47\%}$

Table 4: **Next-frame video generation on Self-Forcing (5 s) and LongLive (30 s).** KV reports cache size (GB) with reduction vs. BF16 in subscript. On generation quality, **bold** and underline indicate best and second-best, respectively.

Model	Method	Prec.	VBench [8] ↑						Opus4.7-VL ↑			KV (GB)↓
			Subject Consistency	Background Consistency	Motion Smoothness	Dynamic Degree	Aesthetic Quality	Total Quality	Visual	Dynamic	Text	
Self-Forcing 1.3B (5-second)	Baseline	BF16	93.57	94.07	98.43	75.00	60.07	82.94	3.81	3.10	3.62	$67.49_{\text{bs=12}}$
	FlexGen	INT6	93.52	94.11	98.44	76.00	60.02	82.98	3.79	3.15	**3.73**	$27.95_{-58.59\%}$
	KIVI	INT6	93.58	94.12	**98.49**	73.00	60.07	82.85	3.82	3.14	3.63	$28.22_{-58.18\%}$
	VAR-Q	INT6	**93.96**	**94.76**	**98.49**	**76.50**	**61.25**	**83.34**	**3.89**	**3.17**	3.62	**27.50**$_{-59.25\%}$
	FlexGen	INT4	86.39	85.84	95.40	73.00	57.39	75.91	**3.60**	3.40	3.75	$17.40_{-74.22\%}$
	KIVI	INT4	90.53	91.86	98.29	74.50	**57.64**	80.33	3.54	3.44	**3.76**	$17.68_{-73.81\%}$
	VAR-Q	INT4	**91.28**	**92.55**	**98.35**	**75.50**	57.40	**80.99**	3.55	**3.50**	3.73	**16.95**$_{-74.88\%}$
LongLive 1.3B (30-second)	Baseline	BF16	91.44	93.83	98.94	44.50	60.74	80.92	3.75	3.25	3.69	$64.28_{\text{bs=20}}$
	FlexGen	INT6	**91.33**	93.49	**98.95**	46.50	60.29	80.60	3.78	**3.33**	**3.78**	$26.62_{-58.59\%}$
	KIVI	INT6	91.14	93.45	98.94	43.50	**60.36**	80.65	3.73	3.27	3.70	$26.88_{-58.18\%}$
	VAR-Q	INT6	91.26	**93.57**	98.93	**47.50**	60.17	**80.94**	**3.84**	3.31	3.71	**26.19**$_{-59.25\%}$
	FlexGen	INT4	86.72	90.19	95.83	45.00	55.16	76.59	3.48	3.48	3.77	$16.57_{-74.22\%}$
	KIVI	INT4	85.83	**92.09**	97.80	**49.50**	56.59	78.95	3.48	3.55	3.79	$16.84_{-73.81\%}$
	VAR-Q	INT4	**88.39**	91.97	**98.80**	47.50	**57.11**	**79.60**	**3.54**	**3.57**	**3.80**	**16.15**$_{-74.88\%}$

Self-Forcing and LongLive. At INT6, VAR-Q is indistinguishable from BF16 on VBench Total (83.34 and 80.94 vs. 82.94 and 80.92). Opus4.7-VL ratings on LongLive corroborate this: 3.84 / 3.31 / 3.71 against BF16's 3.75 / 3.25 / 3.69. At INT4, VAR-Q 's gap to BF16 stays within 2 VBench points (1.95 and 1.32), against 2.61 and 1.97 for KIVI and 7.03 and 4.33 for FlexGen; cache savings reach 75%. Together with the VAR result in Section 4.1, these results establish that VAR-Q 's two-axis grouping transfers across both visual AR paradigms.

4.3 Ablation Studies

Grouping rule. Table 5 ablates the two axes of VAR-Q at 4-bit on VAR-d30 (next-scale) and Self-Forcing INT4 (next-frame). On VAR-d30, removing the channel axis collapses generation (FID 12.07, IS 159.52); removing the block axis raises FID by 0.86 (1.97 to 2.83). The channel axis is the dominant factor, the block axis secondary. The trend persists on Self-Forcing: removing the channel axis drops Subject Consistency from 91.22 to 79.17 and Dynamic Degree from 74.00 to 27.00. VAR-Q covers both axes by construction and is therefore the safe paradigm-agnostic default. Among LLM-style alternatives, per-head per-token trails by 0.24 FID on VAR-d30 and 1.04 points on Subject; per-token alone collapses on both backbones with FID of 7.13 and Subject of 78.55.

Table 5: Grouping ablation at INT4 on next-scale VAR-d30 and next-frame Self-Forcing. `VAR-Q` (full) is the proposed method. Shaded rows ablate one axis; the bottom rows compare to LLM-style alternatives. Bold marks the best in each column.

Grouping	VAR-d30 (next-scale)						Self-Forcing (next-frame)				
	FID↓	IS↑	PSNR↑	SSIM↑	LPIPS↓	KV (GB)↓	Subject↑	Dynamic↑	Aesthetic↑	Total↑	KV (GB)↓
`VAR-Q` (full)	**1.97**	**323.67**	**13.54**	**0.312**	**0.437**	10.44$_{-72.06\%}$	**91.28**	**75.50**	**57.40**	**80.99**	16.95
w/o block-axis	2.83	292.13	12.07	0.257	0.538	9.45$_{-74.71\%}$	88.49	69.00	54.51	79.38	16.90
w/o channel-axis	12.07	159.52	10.42	0.198	0.645	**9.34**$_{-75.00\%}$	79.17	27.00	33.86	67.22	**16.87**
Additional baselines (LLM-style groupings):											
Per-token & per-head	2.21	292.12	11.97	0.261	0.545	10.51$_{-71.87\%}$	90.18	67.00	54.96	79.46	17.93
Per-token	7.13	203.50	10.51	0.208	0.635	9.38$_{-74.90\%}$	78.55	25.50	39.51	67.72	16.96

Figure 9: Effect of sub-block ratio r.

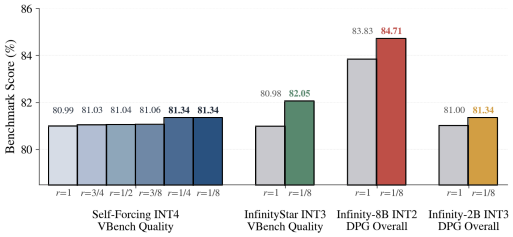

Table 6: Boundary alignment ablation.

Metric	Aligned	Misaligned
Subject consistency ↑	**91.23**	90.80
Dynamic degree ↑	**74.50**	73.50
Aesthetic quality ↑	**57.41**	56.75
VBench total score ↑	**80.91**	80.46

Effect of sub-block ratio r. Beyond the default per-block grouping ($r = 1$), finer sub-blocks improve quality on every backbone we test (Figure 9). On Self-Forcing INT4, VBench Quality rises from 80.99 at $r = 1$ to 81.34 at $r = 1/4$ and saturates at $r = 1/8$. At $r = 1/8$, gains range from $+0.34$ for Infinity-2B INT3 to $+1.07$ for InfinityStar INT3, with Infinity-8B INT2 at $+0.88$. Crucially, r partitions within existing block boundaries, so this refinement adds no detection or calibration cost.

Effect of Boundary Alignment. We define two schemes with exactly matched groupings. *Aligned* places group boundaries on block boundaries; *misaligned* shifts every boundary by half a block. This minimal perturbation reduces the generation quality of Self-Forcing by 0.45 in Table 6 and uniformly degrades all dimensions.

5 Related Work

Visual autoregression has consolidated around two block-wise paradigms: *next-scale* prediction [4, 12, 24], which decodes images coarse-to-fine across resolutions, and *next-frame* prediction [7, 29], which decodes video frame-by-frame. Both emit a block of tokens per autoregressive step, so the KV cache scales with horizon and dominates memory in long-form synthesis. KV-cache compression for language models is mature, spanning low-precision quantization [2, 5, 13, 14, 21, 30], attention-guided eviction [11, 32], and hybrid quantization-sparsification [9]. These methods inherit token-centric assumptions of uniform importance, linear cache growth, and stationary statistics across steps, all of which fail under the block-wise dynamics of visual AR, where magnitudes shift sharply at block boundaries and concentrate on a small set of channels. Extensions to diffusion transformers [15] and multimodal LLMs [25] target adjacent regimes but inherit the same statistics. The closest concurrent work addresses visual AR directly. KV eviction [10, 28] and sparse-attention accelerators [26] reduce *which* entries are retained rather than their precision, and thus compose orthogonally with `VAR-Q`. Concurrent quantization efforts [16, 27] target next-scale image and next-frame video respectively, but neither attributes the cache structure to its architectural sources nor exploits the joint channel-block grouping; `VAR-Q` reuses block boundaries from the generation loop and applies one calibration-free rule across both paradigms.

6 Conclusion

VAR-Q groups KV activations along channel and block axes, reusing boundaries supplied by the generation loop and requiring no calibration. Across nine visual AR models spanning both next-scale and next-frame paradigms, it cuts KV memory by up to 78.5% with negligible quality degradation, enables 6× larger inference batches, and makes large visual autoregressive models more practical to deploy.

Acknowledgments and Disclosure of Funding

This material is based upon work supported by the National Science Foundation under Grants No. 1948201 and No. 2310170. The authors declare no competing interests.

References

[1] 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. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

[2] Daehyeon Baek, Jieun Choi, Jimyoung Son, Kyungmin Bin, Seungbeom Choi, Kihyo Moon, Minsung Jang, and Hyojung Lee. Fireq: Fast int4-fp8 kernel and rope-aware quantization for llm inference acceleration. *arXiv preprint arXiv:2505.20839*, 2025.

[3] Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36:52132–52152, 2023.

[4] Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 15733–15744, 2025.

[5] 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.

[6] Xiwei Hu, Rui Wang, Yixiao Fang, Bin Fu, Pei Cheng, and Gang Yu. Ella: Equip diffusion models with llm for enhanced semantic alignment. *arXiv preprint arXiv:2403.05135*, 2024.

[7] Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025.

[8] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21807–21818, 2024.

[9] Hao Kang, Qingru Zhang, Souvik Kundu, Geonhwa Jeong, Zaoxing Liu, Tushar Krishna, and Tuo Zhao. GEAR: An efficient error reduction framework for KV cache compression in LLM inference. In Mehdi Rezagholizadeh, Peyman Passban, Soheila Samiee, Vahid Partovi Nia, Yu Cheng, Yue Deng, Qun Liu, and Boxing Chen, editors, *Proceedings of The 4th NeurIPS Efficient Natural Language and Speech Processing Workshop*, volume 262 of *Proceedings of Machine Learning Research*, pages 305–321. PMLR, 14 Dec 2024. URL https://proceedings.mlr.press/v262/kang24a.html.

[10] Kunjun Li, Zigeng Chen, Cheng-Yen Yang, and Jenq-Neng Hwang. Memory-efficient visual autoregressive modeling with scale-aware KV cache compression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[11] Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. Snapkv: LLM knows what you are looking for before generation. *Advances in Neural Information Processing Systems*, 37:22947–22970, 2024.

[12] Jinlai Liu, Jian Han, Bin Yan, Hui Wu, Fengda Zhu, Xing Wang, Yi Jiang, Bingyue Peng, and Zehuan Yuan. Infinitystar: Unified spacetime autoregressive modeling for visual generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[13] 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. *arXiv preprint arXiv:2405.16406*, 2024.

[14] 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*, 2024.

[15] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Deepcache: Accelerating diffusion models for free. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15762–15772, 2024.

[16] Jaehyeon Moon and Bumsub Ham. Shift-and-sum quantization for visual autoregressive models. In *The Fourteenth International Conference on Learning Representations*, 2026.

[17] 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.

[18] Ying Sheng, Lianmin Zheng, Binhang Yuan, Zhuohan Li, Max Ryabinin, Beidi Chen, Percy Liang, Christopher Ré, Ion Stoica, and Ce Zhang. Flexgen: High-throughput generative inference of large language models with a single gpu. In *International Conference on Machine Learning*, pages 31094–31116. PMLR, 2023.

[19] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

[20] Zunhai Su, Wang Shen, Linge Li, Zhe Chen, Hanyu Wei, Huangqi Yu, and Kehong Yuan. Akvq-vl: Attention-aware kv cache adaptive 2-bit quantization for vision-language models. In *2025 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2025.

[21] Zunhai Su, Hanyu Wei, Zhe Chen, Wang Shen, Linge Li, Huangqi Yu, and Kehong Yuan. Rotatekv: accurate and robust 2-bit kv cache quantization for llms via outlier-aware adaptive rotations. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, IJCAI '25, 2025. ISBN 978-1-956792-06-5. doi: 10.24963/ijcai.2025/690. URL `https://doi.org/10.24963/ijcai.2025/690`.

[22] Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. Massive activations in large language models. In *First Conference on Language Modeling (COLM)*, 2024.

[23] Qian Tao, Wenyuan Yu, and Jingren Zhou. Asymkv: Enabling 1-bit quantization of kv cache with layer-wise asymmetric quantization configurations. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2316–2328, 2025.

[24] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

[25] Zhongwei Wan, Ziang Wu, Che Liu, Jinfa Huang, Zhihong Zhu, Peng Jin, Longyue Wang, and Li Yuan. Look-m: Look-once optimization in kv cache for efficient multimodal long-context inference. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4065–4078, 2024.

[26] Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, Jianfei Chen, Ion Stoica, Kurt Keutzer, and Song Han. Sparse video-gen: Accelerating video diffusion transformers with spatial-temporal sparsity. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 68208–68224. PMLR, 13–19 Jul 2025. URL `https://proceedings.mlr.press/v267/xi25c.html`.

[27] 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.

[28] Boxun Xu, Yu Wang, Zihu Wang, and Peng Li. Ams-kv: Adaptive kv caching in multi-scale visual autoregressive transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 27206–27214, 2026.

[29] 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*, 2025.

[30] Dingyu Yao et al. VecInfer: Efficient LLM inference with low-bit KV cache via outlier-suppressed vector quantization. *arXiv preprint arXiv:2510.06175*, 2025.

[31] Mengxia Yu, De Wang, Qi Shan, Colorado J Reed, and Alvin Wan. The super weight in large language models. *arXiv preprint arXiv:2411.07191*, 2024.

[32] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710, 2023.

[33] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

A Approximate ℓ_2 Invariance of Paired Channel Maxima under RoPE

Within one block of T tokens, let pair (c_t, d_t) at token t have pre-RoPE amplitude $M_t = \sqrt{c_t^2 + d_t^2}$, with order statistics $M_{(1)} \geq M_{(2)} \geq \cdots \geq M_{(T)}$. After RoPE rotation by a shared angle θ, write $a_t = M_t \cos(\theta + \phi_t)$ and $b_t = M_t \sin(\theta + \phi_t)$, where ϕ_t is the per-token pre-RoPE angle. The pair of channel maxes satisfies

$$M_{(1)}^2 \;\leq\; \|a\|_\infty^2 + \|b\|_\infty^2 \;\leq\; M_{(1)}^2 + M_{(2)}^2. \tag{4}$$

Lower bound. Let $t^\star = \arg\max_t M_t$. Per-token RoPE invariance gives $a_{t^\star}^2 + b_{t^\star}^2 = M_{(1)}^2$. Since $\|a\|_\infty \geq |a_{t^\star}|$ and $\|b\|_\infty \geq |b_{t^\star}|$, the sum is at least $M_{(1)}^2$.

Upper bound. Let $t_a = \arg\max_t |a_t|$ and $t_b = \arg\max_t |b_t|$. Per-token, $|a_t|, |b_t| \leq M_t$, so $\|a\|_\infty^2 \leq M_{t_a}^2$ and $\|b\|_\infty^2 \leq M_{t_b}^2$. If $t_a = t_b$, the sum equals $M_{t_a}^2 \leq M_{(1)}^2$. If $t_a \neq t_b$, at least one of them is not $t^\star$, so $\min(M_{t_a}, M_{t_b}) \leq M_{(2)}$ and the sum is at most $M_{(1)}^2 + M_{(2)}^2$.

Bounded variation. For $M_{(1)} > 0$, let $\rho = M_{(2)}/M_{(1)} \leq 1$. The bounds above place the ℓ_2 norm of the paired channel maxima in $[M_{(1)}, M_{(1)}\sqrt{1+\rho^2}]$ for every rotation angle. Thus, even when the two largest amplitudes are comparable, the ratio between the larger and smaller norms at any two angles is at most $\sqrt{1+\rho^2} \leq \sqrt{2}$. If $M_{(1)} = 0$, all values are zero and invariance is exact.

Approximate invariance. When the largest pre-RoPE amplitude dominates within the block, $\rho \ll 1$, the interval narrows and the paired channel-max norm is approximately invariant under RoPE. This dominance condition is sufficient for the envelope-flat behavior illustrated in Fig. 7(b); heavy-tailed amplitudes alone do not guarantee it.

B Asymmetric Roles of K and V Across Modalities

B.1 K/V asymmetry between image and video AR

The joint result above masks a per-tensor split: **video models are more sensitive to $\mathbf{K}$ outliers, while image models are more sensitive to $\mathbf{V}$ outliers**. Mechanistically, $\mathbf{K}$ determines which past tokens are attended to and $\mathbf{V}$ determines what content is retrieved. The split therefore suggests that video generation places a heavier demand on precise temporal addressing into the cache, whereas image generation tolerates coarser addressing but depends more on the fidelity of the content retrieved at each cached position. The same per-tensor ablation on LLMs reproduces an LLM-side $\mathbf{K}$-dominance consistent with prior reports [14, 23], placing video AR closer to the LLM regime and image AR farther from it.

This appendix provides per-tensor evidence for the $\mathbf{K/V}$ asymmetry discussed in the main paper, separating $\mathbf{K}$-only and $\mathbf{V}$-only ablations across image and video AR, and extending the comparison to LLaMA-3-8B-Instruct. We follow the channel-wise ablation protocol of the main paper but mask $\mathbf{K}$ and $\mathbf{V}$ independently. Channels are ranked by per-token absolute maximum aggregated over a calibration set; the top, bottom, or random $p\%$ are zeroed at inference, with $p \in \{0.1, 0.5, 3, 5\}$. Four visual AR backbones (Infinity-2B, Infinity-8B, LongLive-30s, InfinityStar-720p) are evaluated with the perceptual and VBench metrics used in the main paper, and LLaMA-3-8B-Instruct is evaluated on WikiText-2 perplexity (4096-token chunks), MMLU (200 sampled questions), and GSM8K (50 sampled questions, chat-template greedy generation). Random and inlier ablation arms remain within metric-level noise of the unmasked baseline across all backbones and ratios, confirming that the observed degradation is selective to high-magnitude channels rather than to channel count. One measurement caveat applies to video: the VBench dimensions *dynamic_degree* and *temporal_flickering* register increases under $\mathbf{K}$-outlier ablation that do not reflect genuine motion, since the per-frame chroma drift induced by $\mathbf{K}$-side collapse is misclassified as motion by the underlying classifiers; we therefore rely on *imaging_quality* (MUSIQ) and *overall_consistency* (ViCLIP) as the most reliable VBench dimensions for this perturbation regime.

B.2 Extension to LLaMA-3-8B

The top two rows of Figure 10 show that image and video AR sit on opposite sides of the $\mathbf{K/V}$ axis. Image AR is $\mathbf{V}$-dominant: on Infinity-2B at $p = 5\%$, $\mathbf{V}$-only ablation drives ΔLPIPS to 0.20

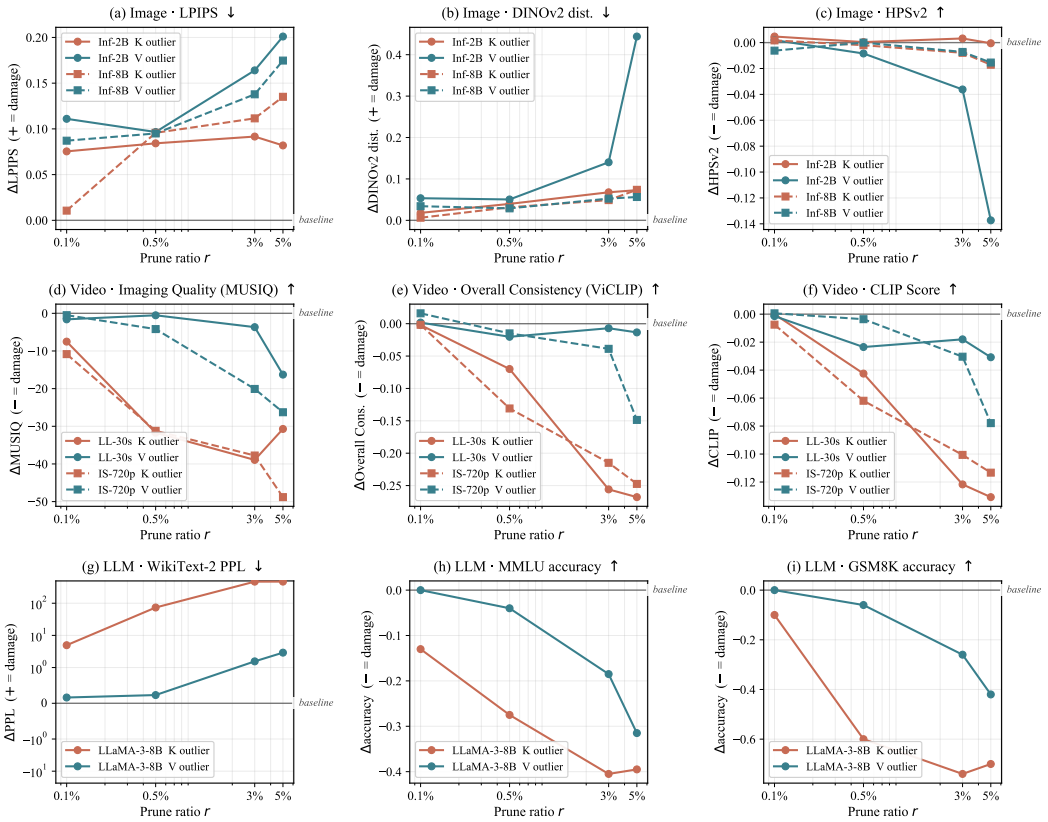

Figure 10: **K**-only (terracotta) versus **V**-only (teal blue) outlier ablation. **Top row**: image AR backbones on perceptual and aesthetic metrics. **Middle row**: video AR backbones on perceptual and VBench-MUSIQ metrics. **Bottom row**: LLaMA-3-8B-Instruct on language benchmarks (PPL on log scale). All metrics are oriented so that higher values indicate worse generation. The dominant color flips between image and video AR.

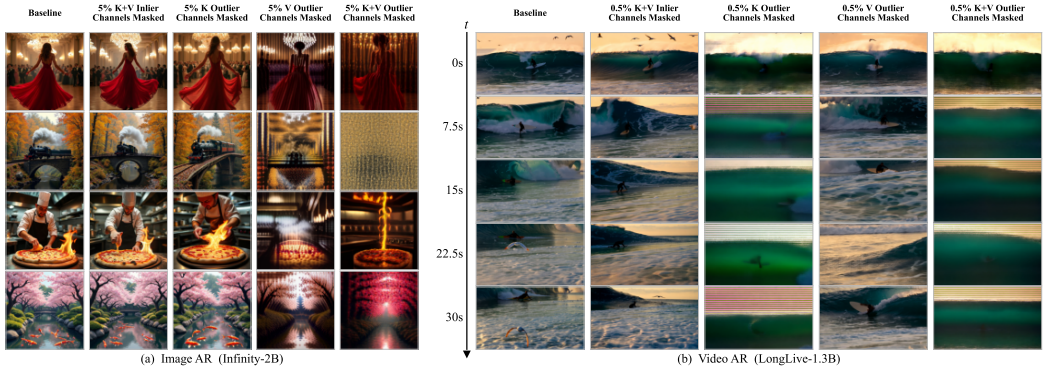

Figure 11: Generation under **K**-only, **V**-only, and joint **K**+**V** outlier masking. **(a)** Infinity-2B at $p = 5\%$: **V**-only masking collapses outputs into texture or symmetric patterns, while **K**-only masking leaves outputs visually close to the baseline. **(b)** LongLive-1.3B at $p = 0.5\%$: **K**-only masking collapses frames into solid color and chroma stripes, while **V**-only masking preserves scene layout with degraded fine detail. Inlier masking matches the baseline in both cases.

against 0.08 for **K**-only, and the LAION aesthetic score drops by 1.3 under **V**-only against near-zero under **K**-only. Video AR reverses the direction: on LongLive-30s at $p = 5\%$, **K**-only ablation drives $\Delta\text{LPIPS}_{\text{mid}}$ to 0.40 against 0.14 for **V**-only, and the same direction holds on InfinityStar-720p. The image-side **V**-dominance is consistent across both Infinity backbones and the video-side **K**-dominance across both video backbones, indicating that the asymmetry tracks the modality rather than per-model idiosyncrasy. Figure 11 renders the same contrast at the sample level: on Infinity-2B, **V**-only masking eliminates recognizable scene content while **K**-only masking is nearly invisible; on LongLive-1.3B, **K**-only masking collapses frames into uniform color while **V**-only masking preserves scene layout with degraded texture. The joint **K**+**V** mask in both cases tracks the more damaging single-tensor mask, consistent with the dominant tensor carrying the load-bearing signal.

The bottom row of Figure 10 extends the same protocol to LLaMA-3-8B-Instruct. The model behaves like video AR in direction but more sharply in magnitude: at $p = 0.5\%$, **K**-only ablation increases WikiText-2 perplexity from 8.4 to 82.4 ($\sim 10\times$), while **V**-only ablation leaves it at 11.5 ($1.4\times$); MMLU and GSM8K accuracy degrade in the same direction. Figure 12 summarizes the damage ratio $\Delta_{\mathbf{K}}/\Delta_{\mathbf{V}}$ at $p = 0.5\%$: image AR sits at or below $1\times$, video AR ranges from $2.5\times$ to $61\times$, and LLaMA-3-8B reaches $320\times$ on perplexity. The LLM observation aligns with prior reports of **K**-side outlier dominance in LLM KV caches [14, 23], placing video AR closer to the LLM regime while leaving image AR as the singular setting in which **V** outliers carry the bulk of the load.

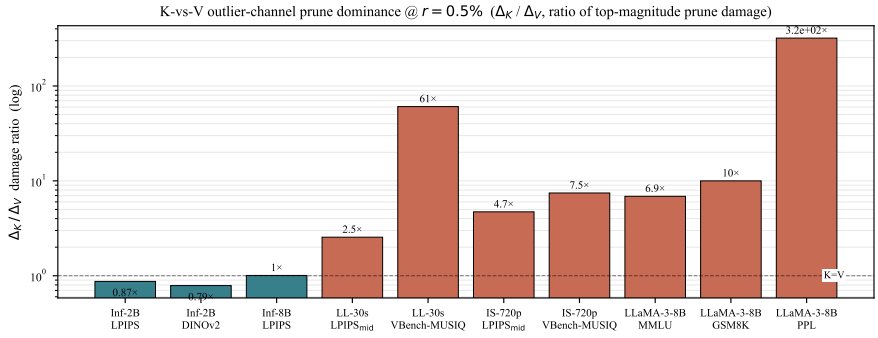

Figure 12: $\Delta_{\mathbf{K}}/\Delta_{\mathbf{V}}$ damage ratio at $p = 0.5\%$ (log scale). Bars below the dashed **K**=**V** line are **V**-dominant; bars above are **K**-dominant. Image AR sits at or below $1\times$, video AR ranges from $2.5\times$ to $61\times$, and LLaMA-3-8B reaches $320\times$ on perplexity.

C Persistence Across Different Prompts

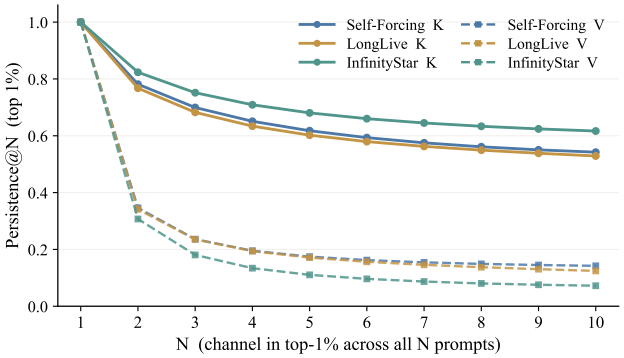

Figure 13: Outlier persistence as the number of prompts increases

As shown in Figure 13, the shapes of the K and V curves differ qualitatively, not just in magnitude. V (dashed) collapses on the first added prompt: even a single second prompt removes about two-thirds of the outlier channels (Persistence@2 $= 0.31$ to 0.35 across the three backbones), and the curve continues to decay toward 0.07 to 0.14. K (solid) instead bends gently, retaining 0.77 to 0.82

at $N = 2$ and asymptoting at 0.53 to 0.62. The K-V gap is already $2.3\times$ at $N = 2$ and widens monotonically with N, so the $N = 10$ cross-section reported as Persistence@10 is the rightmost point of two curves whose decay rates differ throughout. The plateau shape on K is the signature of an architectural core: most outlier channels survive every additional prompt, consistent with their being selected by the static W_K, γ_K, and RoPE-pair chain (Section 3.3). The crash on V is the signature of input-driven outliers: each new prompt is a fresh sample of token amplitudes, and the V channel identity tracks those samples.

D Additional Experiments and Deployment Analysis

We extend the evaluation of VAR-Q to deployment efficiency, metadata overhead, and comparisons with visual-AR quantization and sparse attention. We also combine VAR-Q with weight and activation quantization. Deployment measurements use an NVIDIA A100 80 GB; each experiment specifies its workload and memory accounting.

D.1 End-to-End Latency and Batch Capacity

We measure full-generation latency using CUDA 12.2, PyTorch, and FlashAttention-2, with three warmup runs followed by ten timed runs. Each model uses its native sequence and block lengths. We first evaluate batch-one latency, then isolate the effect of kernel fusion and examine batch scaling. All end-to-end timings include quantization, packing, unpacking, and dequantization. We denote batch size by B_{batch} to distinguish it from the block count B in the main text.

Table 7: End-to-end generation latency (seconds, ↓) at batch size 1. Columns specify KV-cache precision. Timings include quantization, packing, unpacking, and dequantization; each entry averages ten runs after three warmups.

Model / output	BF16	VAR-Q KV-cache precision			
		INT8	INT4	INT3	INT2
Infinity-8B / image	3.18	3.63	3.60	3.71	3.63
Self-Forcing / 5 s	11.87	12.39	12.24	12.27	12.11
LongLive / 30 s	78.21	79.30	78.34	79.30	78.34
InfinityStar-480p / 5 s	53.12	55.97	56.13	56.24	56.14

Batch-one latency. Table 7 shows that INT4 increases latency from 3.18 to 3.60 s on Infinity-8B, from 11.87 to 12.24 s on Self-Forcing, and from 78.21 to 78.34 s on LongLive. InfinityStar-480p takes 56.13 s versus 53.12 s for BF16. The overhead varies across workloads, with the smallest relative change on LongLive; cache compression does not produce a batch-one speedup in these measurements.

Kernel fusion. Quantization and packing run in Triton when new KV blocks are written. A modified FlashAttention-2 CUDA kernel reads the packed blocks, unpacks values in registers/shared memory, and applies scales during attention. This avoids materializing a full BF16 cache, as required by the unfused reference. Table 8 compares the two paths in an Infinity-8B microbenchmark with a common BF16 reference of 3.164 s.

Table 8: Effect of kernel fusion on Infinity-8B at batch size 1. Attention-path timings include unpacking and dequantization. BF16 is the common reference for both implementations. All latencies are in milliseconds (↓). Bold/underline mark the best/second-best latency within each fused/unfused pair at the same precision; ties are both bold.

Measurement	Implementation	BF16	VAR-Q INT8	VAR-Q INT4
Attention	Unfused	**1427**	<u>1951</u>	<u>1999</u>
	Fused	**1427**	**1603**	**1583**
End-to-end	Unfused	**3164**	<u>3973</u>	<u>4012</u>
	Fused	**3164**	**3627**	**3596**

Table 9: Latency profile for LongLive 30-second generation at batch size 1. All latencies are in milliseconds (↓). A dash denotes an unreported component. Component measurements may overlap when operations are fused and should not be summed to obtain end-to-end latency.

Operation	BF16	VAR-Q INT8	VAR-Q INT4
End-to-end generation	78210	79296	78340
Attention	—	20087	20191
Cache read/write	—	4379	4429
Quantization and packing	—	2547	3024
Unpacking and dequantization	—	1698	1646

Fusion reduces Infinity-8B end-to-end latency by 8.7% at INT8 and 10.4% at INT4 relative to the unfused path, although both remain above BF16. The LongLive profile in Table 9 shows end-to-end overheads of 1.4% and 0.2%, respectively. Quantization and packing account for 2.547/3.024 s at INT8/INT4, and unpacking and dequantization for 1.698/1.646 s. These component timings can overlap with fused operations and are not additive wall-clock overheads.

Batch scaling. Table 10 shows where reduced cache storage translates into additional batch capacity. Infinity-8B runs at batch 16 with INT2, whereas BF16 runs out of memory at batch 4. Self-Forcing extends from batch 8 to 16 at INT4, and LongLive from batch 4 to 8 at INT3. Larger batches do not imply higher throughput: using $60B_{\text{batch}}/t$ samples per minute, where t is the measured latency in seconds per batch, Infinity-8B reaches 20.91 images/min with BF16 at batch 2 and 18.06 images/min with INT2 at batch 16. The benefit in this sweep is the ability to run larger batches within device memory.

Table 10: End-to-end generation latency (seconds per batch, ↓) across batch sizes. OOM denotes out of memory. A dash denotes an entry not used in this comparison.

Model	Precision	Batch size 2	Batch size 4	Batch size 8	Batch size 16
Infinity-8B	BF16	5.74	OOM	OOM	OOM
	VAR-Q INT8	7.24	13.67	OOM	OOM
	VAR-Q INT4	7.41	13.93	27.05	OOM
	VAR-Q INT3	7.80	14.40	28.10	—
	VAR-Q INT2	7.50	13.86	26.76	53.17
Self-Forcing	BF16	27.40	48.60	91.00	OOM
	VAR-Q INT8	32.60	61.80	120.40	OOM
	VAR-Q INT4	34.60	65.50	127.60	252.30
	VAR-Q INT3	35.00	66.20	129.00	255.90
	VAR-Q INT2	34.50	64.90	126.50	249.90
LongLive	BF16	149.43	288.66	OOM	—
	VAR-Q INT8	152.70	296.35	OOM	—
	VAR-Q INT4	151.92	294.49	OOM	—
	VAR-Q INT3	151.35	292.99	575.73	—
	VAR-Q INT2	149.49	289.53	569.55	—

Memory accounting. The 78.5% reduction reported for InfinityStar INT3 is a KV-cache reduction, from 55.87 to 12.03 GB, rather than a reduction in total device memory. The additional LongLive measurements give KV-cache reductions of 49.9%, 74.9%, 79.6%, and 87.4% at INT8/4/3/2. The achieved saving therefore depends on the model configuration and precision.

D.2 Metadata Overhead and Effective Compression

The measured configurations use signed symmetric quantization with one BF16 scale for each (channel, block) group of K and V, without a zero-point tensor. Across both caches and all layers, N_{scale} stored scales require $2N_{\text{scale}}$ bytes. We account for this overhead using $2N_{\text{scale}}/(M_{\text{packed}}+2N_{\text{scale}})$, where M_{packed} includes word padding. Shape and packing descriptors are separate from the resident GPU scale tensors.

Table 11: Storage occupied by BF16 quantization scales. Metadata fractions use the stored quantized KV cache as the denominator; absolute sizes depend on the model configuration.

Model	Scale storage (MB) ↓	Metadata fraction (%) ↓	
		VAR-Q INT8	VAR-Q INT4
Infinity-8B	13.80	0.37	0.74
InfinityStar	0.29	0.12	0.24
Self-Forcing	3.90	0.13	0.26
LongLive	2.20	0.13	0.26

Table 12: KV-cache storage on Infinity-8B, including packing and metadata. Compression is BF16 cache bytes divided by stored quantized cache bytes. KIVI stores scales and zero-points; VAR-Q stores scales. This separate audit gives a VAR-Q INT4 metadata fraction of 0.75%, compared with 0.74% in Table 11. Bold/underline mark the best/second-best method at each precision: lower metadata fraction and higher compression.

Precision	Metadata fraction (%) ↓		Compression ratio ↑	
	VAR-Q	KIVI	VAR-Q	KIVI
INT4	**0.75**	<u>4.88</u>	**3.97**×	<u>3.81</u>×
INT3	**0.94**	<u>5.93</u>	**4.95**×	<u>4.63</u>×
INT2	**1.50**	<u>9.29</u>	**7.88**×	<u>7.26</u>×

Table 11 shows that scales occupy at most 0.74% of the stored cache at INT4 across the measured configurations. The separate Infinity-8B audit in Table 12 reports 0.75% for VAR-Q, versus 4.88% for KIVI, which also stores zero-points. Including packing and metadata, VAR-Q achieves compression ratios of $3.97\times$, $4.95\times$, and $7.88\times$ at INT4/3/2, compared with $3.81\times$, $4.63\times$, and $7.26\times$ for KIVI. At INT3, ten values occupy each 32-bit word; the unused bits explain why effective compression falls below the ideal $16/3$ ratio even with small scale overhead.

D.3 Comparison with Visual-AR KV-Cache Quantization

We compare VAR-Q with Quant VideoGen (QVG) [27], a KV-cache quantization framework developed for autoregressive video diffusion models. The comparison covers next-scale image generation on Infinity-8B and next-frame video generation on Self-Forcing. We use QVG's released implementation and default configuration, with the residual setting recorded as BF16 for all QVG results reported here. All methods in Table 13 use the same GenEval evaluation protocol; these additional runs are reported separately from the main-table results.

Table 13: Infinity-8B comparison at INT2. GenEval reports the overall score on a 0–100 scale; latency is end-to-end. Both methods use nominal INT2 precision; measured storage includes the cache retained by each implementation. Bold/underline mark the best/second-best result at INT2 for each metric.

Method	Precision	GenEval ↑	Latency (s) ↓	KV storage (GB) ↓
QVG†	INT2	<u>70.63</u>	<u>4.90</u>	<u>1.95</u>
VAR-Q	INT2	**75.75**	**3.63**	**0.94**

†QVG: residual = BF16 for all reported precisions.

Infinity-8B. At INT2, VAR-Q reaches 75.75 GenEval versus QVG's 70.63, a 5.12-point advantage (Table 13). It also reduces KV storage from 1.95 to 0.94 GB and end-to-end latency from 4.90 to 3.63 s ($1.35\times$ speedup). The quality advantage is therefore accompanied by lower measured storage and latency at the same nominal bit-width.

Table 14: Self-Forcing comparison at INT4 and INT2 using QVG and VAR-Q. Latency is measured on a 180-latent-frame workload. The latency workload differs from the five-second clips in Table 7. Bold/underline mark the best/second-best method within each precision for each metric; ties are both bold.

Metric	INT4		INT2	
	QVG†	VAR-Q	QVG†	VAR-Q
Subject consistency ↑	<u>93.31</u>	**93.89**	<u>86.88</u>	**88.30**
Background consistency ↑	<u>94.12</u>	**94.35**	<u>89.36</u>	**91.12**
Motion smoothness ↑	**98.49**	<u>98.48</u>	<u>98.47</u>	**98.49**
Aesthetic quality ↑	<u>60.03</u>	**60.56**	<u>51.49</u>	**55.04**
Imaging quality ↑	<u>69.90</u>	**70.60**	<u>58.57</u>	**60.90**
VBench total score ↑	<u>82.67</u>	**82.93**	<u>75.55</u>	**77.67**
End-to-end latency (s) ↓	<u>487.66</u>	**334.21**	<u>452.70</u>	**324.71**

†QVG: residual = BF16 for all reported precisions.

Self-Forcing. At INT4, VAR-Q improves VBench Total from 82.67 to 82.93, with higher scores on all reported dimensions except Motion Smoothness (98.48 vs. 98.49). The gap widens at INT2: VAR-Q reaches 77.67 versus QVG's 75.55 and leads on all six metrics (Table 14). On the 180-latent-frame workload, VAR-Q takes 334.21 s at INT4 and 324.71 s at INT2, yielding $1.46\times$ and $1.39\times$ speedups over QVG. Together with the Infinity-8B results, this shows higher aggregate quality and lower measured latency on both tested backbones.

D.4 Comparison and Composition with Sparse Attention

Sparse attention reduces the tokens or blocks accessed by attention; VAR-Q reduces the precision of the cached keys and values. We compare their quality at matched KV-cache storage and then evaluate whether VAR-Q preserves quality when applied on top of sparse attention.

Comparison with Sparse Attention at Matched Storage Budget. Tables 15 and 16 compare VAR-Q with sliding-window and streaming attention on Infinity-8B and Self-Forcing. We match realized KV-cache storage reductions relative to dense BF16. For sparse attention, this requires omitting or evicting unneeded tokens: reducing attention computation alone does not reduce the resident cache.

Table 15: Infinity-8B at matched KV-cache storage reductions. Dense BF16 is the reference. GenEval overall scores are shown on a 0–100 scale. Bold/underline mark the best/second-best GenEval score among methods at the matched storage budget; dense BF16 is a reference.

Method	KV-storage reduction (%) ↑	GenEval ↑
Dense BF16	0.0	83.56
Streaming	74.8	<u>82.85</u>
Sliding window	74.8	82.31
VAR-Q INT4	74.8	**83.47**

Table 16: Self-Forcing at a matched 74.9% KV-cache storage reduction. We compare sliding-window attention, streaming attention, and VAR-Q INT4; dense BF16 (0% reduction) provides the reference. VBench scores are shown on a 0–100 scale. Bold/underline mark the best/second-best score at the matched storage budget, excluding the dense BF16 reference; ties are both bold.

Metric ↑	Dense BF16	Sliding window	Streaming	VAR-Q INT4
Subject consistency	93.57	91.88	<u>93.34</u>	**93.89**
Background consistency	93.87	<u>93.00</u>	90.23	**94.35**
Motion smoothness	98.43	98.17	**98.52**	<u>98.48</u>
Aesthetic quality	60.07	59.64	**60.66**	<u>60.56</u>
Imaging quality	69.94	70.40	**70.93**	<u>70.60</u>
VBench total score	82.87	81.81	<u>82.67</u>	**82.93**

On Infinity-8B, VAR-Q INT4 stays within 0.09 GenEval points of BF16 at a 74.8% cache reduction (83.47 vs. 83.56), ahead of streaming (82.85) and sliding-window attention (82.31). On Self-Forcing, VAR-Q INT4 achieves a VBench Total of 82.93 at 74.9% cache reduction, 0.26 points above streaming. Streaming retains higher Motion Smoothness, Aesthetic Quality, and Imaging Quality. At these storage budgets, retaining the cache at reduced precision gives higher aggregate quality than the tested token-retention policies.

Composition with sparse attention. We next apply VAR-Q to the caches used by sliding-window, streaming, and block-sparse attention (BSA), with BSA using row-wise top-K retention of 25%. Tables 17 and 18 hold the attention policy fixed within each group, and compare BF16 with quantized KV storage to isolate the additional effect of quantization.

Table 17: Self-Forcing with sparse attention and KV-cache quantization. Each pair holds the attention policy fixed; all INT4 and INT8 columns use VAR-Q KV-cache quantization. VBench scores are shown on a 0–100 scale. Within each policy, bold and underlined values denote the best and second-best scores.

	Sliding window		Streaming		Block-sparse	
Metric ↑	BF16	INT4	BF16	INT8	BF16	INT8
Subject consistency	**93.83**	<u>93.34</u>	**93.83**	<u>93.77</u>	**91.14**	<u>91.07</u>
Background consistency	**94.35**	<u>94.12</u>	**94.36**	<u>94.34</u>	**92.83**	**92.83**
Motion smoothness	**98.49**	**98.49**	**98.49**	<u>98.48</u>	<u>98.13</u>	**98.14**
Aesthetic quality	**60.10**	<u>60.03</u>	**60.71**	<u>60.68</u>	<u>59.04</u>	**59.14**
Imaging quality	**70.59**	<u>70.01</u>	<u>70.58</u>	**70.62**	**69.72**	<u>69.70</u>
VBench total score	**82.95**	<u>82.74</u>	**82.99**	<u>82.98</u>	<u>79.67</u>	**79.81**

Table 18: Infinity-8B with sparse attention and KV-cache quantization. Each row holds the attention policy fixed and compares BF16 with VAR-Q INT4 KV-cache storage. GenEval overall scores are shown on a 0–100 scale (↑). Within each policy, bold and underlined values denote the best and second-best scores.

Attention policy	BF16	VAR-Q INT4
Sliding window	<u>82.03</u>	**82.25**
Streaming attention	<u>81.59</u>	**81.69**
Block-sparse attention	**81.48**	<u>81.47</u>

On Self-Forcing, adding VAR-Q changes VBench Total from 82.95 to 82.74 for sliding-window INT4, from 82.99 to 82.98 for streaming INT8, and from 79.67 to 79.81 for BSA INT8. On Infinity-8B, all three INT4 compositions stay within 0.22 GenEval points of their sparse BF16 references, including BSA at INT4 (81.47 vs. 81.48). Across the six configurations reported in these two tables,

the aggregate-score changes range from -0.21 to $+0.22$ points. These results support combining VAR-Q with the tested sparse policies: token retention controls how much history is stored, while quantization controls the bytes used to represent it. These measurements establish quality compatibility for the tested combinations; combined memory savings and runtime are not reported here.

D.5 Composition with Weight and Activation Quantization

We combine VAR-Q with GPTQ W4A8 to evaluate KV-cache compression when weights and activations are already quantized. Tables 19 and 20 separate the effect of W4A8 from the additional effect of quantizing the KV cache and report total memory. These measurements use configurations separate from the batch-capacity sweep in Table 10.

Table 19: Joint weight, activation, and KV-cache quantization on Infinity-8B. GenEval scores are shown on a 0–100 scale. Total memory is measured at batch size 4. Bold and underlined values denote the best and second-best results in each row.

Metric	Full BF16	W4A8 only	W4A8 + VAR-Q			
			INT8	INT4	INT3	INT2
GenEval overall score ↑	**83.56**	78.89	79.92	<u>80.54</u>	80.21	68.44
Single object ↑	**100.00**	<u>98.75</u>	96.25	<u>98.75</u>	97.50	96.25
Two objects ↑	**96.94**	<u>95.92</u>	90.82	93.88	94.90	75.51
Counting ↑	<u>72.29</u>	63.86	**74.70**	69.88	71.08	49.40
Colors ↑	**91.49**	<u>87.23</u>	86.17	86.17	86.17	84.04
Position ↑	**65.66**	59.60	58.59	57.58	<u>60.61</u>	46.46
Color attribution ↑	<u>75.00</u>	68.00	73.00	**77.00**	71.00	59.00
Total memory (GB) ↓	76.5	69.5	50.2	37.1	<u>33.8</u>	**30.5**

W4A8 denotes 4-bit weights and 8-bit activations using the evaluated GPTQ configuration; INT8/4/3/2 in the column headings refers to VAR-Q KV-cache precision. Full BF16 is the unquantized reference; W4A8 only isolates the additional effect of KV quantization.

Infinity-8B. Adding VAR-Q INT4 to W4A8 reduces total memory from 69.5 to 37.1 GB, a further 46.6% reduction. GenEval is 80.54 versus 78.89 for W4A8 alone and 83.56 for full BF16. The individual dimensions move in both directions: Counting and Color Attribution increase, while Two Objects, Colors, and Position decrease. INT2 lowers memory further to 30.5 GB but reduces GenEval to 68.44, making the quality cost of aggressive joint quantization visible.

Table 20: Joint weight, activation, and KV-cache quantization on Self-Forcing. VBench scores are shown on a 0–100 scale. Total memory is measured at batch size 10. Bold and underlined values denote the best and second-best results in each row.

Metric	Full BF16	W4A8 only	W4A8 + VAR-Q	
			INT8	INT4
Subject consistency ↑	93.57	95.13	**95.92**	<u>95.82</u>
Background consistency ↑	93.87	92.99	**94.88**	<u>94.82</u>
Motion smoothness ↑	98.43	98.99	**99.14**	<u>99.09</u>
Aesthetic quality ↑	**60.07**	55.43	<u>56.44</u>	55.72
Imaging quality ↑	**69.94**	64.86	<u>65.64</u>	64.41
VBench total score ↑	**82.87**	78.53	<u>78.95</u>	78.51
Total memory (GB) ↓	82.4[†]	79.4	<u>48.5</u>	**33.5**

W4A8 denotes 4-bit weights and 8-bit activations using the evaluated GPTQ configuration. The INT8 and INT4 columns additionally apply VAR-Q to the KV cache. [†]The 82.4 GB BF16 entry was recorded as out of memory at batch size 10; the accompanying quality scores are reference scores and do not establish a completed run at that batch size.

Self-Forcing. Adding `VAR-Q` INT4 reduces total memory from 79.4 to 33.5 GB, a further 57.8% reduction, while VBench Total stays close to the W4A8 baseline (78.51 vs. 78.53). Both remain below full BF16 (82.87). Thus, KV-cache quantization provides substantial additional memory savings on top of W4A8 while largely preserving the quality of the already-quantized model in this configuration.

E Prompt Template for VLM-based Video Evaluation

Evaluation Prompt. For each video, we partition the timeline into 12 equal segments and sample one frame at the midpoint of each segment, i.e., at relative timestamps $(i+0.5)/12$ for $i = 0, \ldots, 11$. This produces samples spanning 4.17% to 95.83% of the video while excluding boundary regions, where transition artifacts are more common. Frames are extracted with `ffmpeg`, resized to 360 pixels in height with aspect ratio preserved, and stored as JPEG images using `-q:v 3`. To ensure a fair comparison, we keep the 12 sampling locations fixed across all quantization settings for the same prompt, so the VLM judge is exposed to identical temporal evidence. This corresponds to an average spacing of approximately 0.42 seconds for 5-second Self-Forcing videos and 2.5 seconds for 30-second LongLive videos. We provide the exact prompt template used to query the VLM judge for rating text alignment, dynamics, and visual quality.

```
You are a video quality evaluation expert. Below are 12 frames uniformly sampled
from a generated video, plus the text prompt that was used to produce it. Rate
the video on three dimensions (1-5 integer each):

1. Text Alignment -- consistency between video content and the text description.
   - 1: Completely Irrelevant. 2: Mostly Mismatched. 3: Partially Matched.
     4: Largely Consistent. 5: Perfectly Aligned.
2. Dynamics -- dynamism/fluidity of the scene (camera, objects, transitions).
   - 1: Static/Disjointed. 2: Mostly Static. 3: Moderately Dynamic.
     4: Largely Dynamic. 5: Highly Dynamic.
3. Visual Quality -- technical execution (clarity, color, composition, artifacts).
   - 1: Very Poor. 2: Poor. 3: Fair. 4: Good. 5: Excellent.

Output STRICTLY in this format (nothing else, no markdown fences):
Text Alignment: <1-5>
Dynamics: <1-5>
Visual Quality: <1-5>

Text Prompt: """{TEXT_PROMPT}"""
Frames (in temporal order): @f00.jpg @f01.jpg ... @f11.jpg
```

F `VAR-Q` System Support

We quantize each new K or V block independently, using one scale per channel and token group, and pack the resulting integers into 32-bit words. During attention, the integers are unpacked and multiplied by their scales. Algorithms 1–4 give these four operations. The fused attention path performs unpacking and rescaling on tiles; runtime measurements are reported in Appendix D.1.

For clarity, $X \in \mathbb{R}^{L \times C}$ denotes one block with L tokens and C channels; batch, head, and layout indices are omitted. The token groups $I_1, \ldots, I_G$ partition this block, with $G = 1$ by default. We use $b \in \{2, 3, 4, 6, 8\}$ bits and signed bounds $(q_-, q_+) = (-2^{b-1}, 2^{b-1} - 1)$. Rounding uses nearest with ties to even. $\epsilon = 10^{-12}$ is a small constant for numerical stability. $\mathrm{StoreScale}$ casts to the scale-storage dtype; division uses compute precision, and reconstruction casts both operands to the output dtype. Scales must be finite and positive. $\mathrm{uint32}$ interprets a stored word as an unsigned bit pattern.

Algorithm 1: Quantize and pack a KV block

Input: block X, token groups $\{I_g\}$, bit-width b
Output: packed integers P, scales S

foreach *group* I_g *and channel* c **do**
  $S_{g,c} \leftarrow \mathrm{StoreScale}\big(\max\big(\max_{t\in I_g} |X_{t,c}|/q_+, \epsilon\big)\big)$
  $Q_{I_g,c} \leftarrow \mathrm{clip}\big(\mathrm{round}(X_{I_g,c}/S_{g,c}), q_-, q_+\big)$
$P \leftarrow \textsc{Pack}(Q, b)$ `// Algorithm 3`
return P, S

Algorithm 2: Unpack and reconstruct a KV block

Input: packed integers P, scales S, token groups $\{I_g\}$, bit-width b, channel count C
Output: reconstructed block $\widetilde{X}$

$Q \leftarrow \textsc{Unpack}(P, b, C)$ `// Algorithm 4`
foreach *group* I_g *and channel* c **do**
  $\widetilde{X}_{I_g,c} \leftarrow Q_{I_g,c}\, S_{g,c}$
return $\widetilde{X}$

Algorithm 3: Pack signed integers into 32-bit words

Input: signed b-bit integers $Q \in \mathbb{Z}^{L\times C}$, bit-width b
Output: 32-bit words $P \in \mathrm{int32}^{L\times W}$

$w \leftarrow \lfloor 32/b \rfloor$; $W \leftarrow \lceil C/w \rceil$
Pad Q with zero channels to width Ww
foreach *token* t *and word* $j = 0, \ldots, W-1$ **do**
  $P_{t,j} \leftarrow \sum_{k=0}^{w-1} (Q_{t,jw+k} \bmod 2^b)\, 2^{bk}$ `// store the 32-bit pattern`
return P

Algorithm 4: Unpack signed integers from 32-bit words

Input: 32-bit words P, bit-width b, original channel count C
Output: signed integers $Q \in \mathbb{Z}^{L\times C}$

$w \leftarrow \lfloor 32/b \rfloor$
foreach *token* t *and channel* $c = 0, \ldots, C-1$ **do**
  $(j, k) \leftarrow (\lfloor c/w \rfloor, c \bmod w)$
  $u \leftarrow \lfloor \mathrm{uint32}(P_{t,j})/2^{bk} \rfloor \bmod 2^b$
  $Q_{t,c} \leftarrow u - 2^b\, \mathbf{1}[u \geq 2^{b-1}]$
return Q

G Pre-RoPE vs. Post-RoPE Quantization

We ablate the quantization point of `VAR-Q` on Self-Forcing at INT6. As shown in Table 21, post-RoPE quantization consistently outperforms pre-RoPE on the metrics most related to temporal behavior and overall generation quality: Subject Consistency, Background Consistency, Dynamic Degree, Aesthetic Quality, and Total Quality. The gain in Dynamic Degree (76.50 vs. 74.50) indicates that post-RoPE quantization better preserves motion cues instead of over-smoothing the generated sequence.

Table 21: Pre-RoPE vs. post-RoPE quantization with `VAR-Q` INT6. Bold/underline: best/second-best quality.

Variant	Prec.	VBench [8] ↑					KV (GB)↓
		Subj.	Bg.	Dyn.	Aes.	Quality	
post-RoPE	INT6	**93.96**	**94.76**	**76.50**	**61.25**	**83.34**	27.50$_{-59.25\%}$
pre-RoPE	INT6	<u>93.63</u>	<u>93.97</u>	<u>74.50</u>	<u>60.00</u>	<u>82.93</u>	27.50$_{-59.25\%}$

Meanwhile, the improvements in Subject and Background Consistency suggest that quantizing the attention-ready representation also stabilizes the appearance of both foreground objects and scene layout across frames. Post-RoPE further improves Aesthetic Quality and Total Quality (83.34 vs. 82.93), showing that the benefit is not limited to motion but also carries over to perceptual quality.

This supports quantizing the final post-RoPE cache, which is directly consumed by attention, rather than the pre-RoPE cache. We therefore use post-RoPE quantization as the default.

H Evaluation Prompts Adapted from MovieGen

For the MovieGen-based evaluation, we use a fixed subset of 200 prompts randomly sampled from the original MovieGenVideoBench prompt set. We adopt this reduced prompt set purely for efficiency, so that all methods can be evaluated under the same protocol with substantially lower inference cost and turnaround time. The sampled 200 prompts are kept identical across all experiments, models, and quantization settings to ensure fair comparison. No manual prompt selection or prompt-specific tuning is applied after sampling.

Table 22: Complete list of the 200 prompts used for MovieGen-based evaluation. The original line number is 1-based in the full MovieGenVideoBench prompt file.

ID	Line	Prompt
0	621	A person rowing a boat through a river of liquid gold, with shimmering banks.
1	117	Over the shoulder shot of a woman running and watching a rocket in the distance.
2	705	A trampoline surface bending under the weight of a person jumping on it, then springing back up as they jump off.
3	338	A woman's face twisted in disgust when she tasted the spoiled food.
4	50	A little man with blocks visiting an art gallery
5	331	A man's eyes filled with tears of frustration after failing the exam.
6	394	a man and a woman in their 20s are dining in a futuristic restaurant materialized out of nanotech and ferrofluids
7	513	A deep underground cave filled with glowing crystals and hidden treasures.
8	172	A spaceship being pulled into a blackhole.
9	98	A middle-aged sad bald man becomes happy as a wig of curly hair and sunglasses fall suddenly on his head.
10	850	A tracking shot following a skateboarder performing tricks down a city street, keeping pace with their fluid movements.
11	906	a toy robot wearing purple overalls and cowboy boots taking a pleasant stroll in Mumbai India during a colorful festival
12	923	a woman wearing blue jeans and a white t shirt taking a pleasant stroll in Mumbai India during a beautiful sunset
13	378	Gold coins are falling out when elevator door opens
14	756	A low-angle view of a majestic lion standing on a rocky outcrop, looking regal and powerful against the horizon.
15	984	an old man wearing blue jeans and a white t shirt taking a pleasant stroll in Antarctica during a colorful festival
16	305	A gardener planting seeds in a garden bed, their hands gently pressing the soil over the seeds.
17	440	A princess blowing out birthday candles on a cake.
18	920	a toy robot wearing a green dress and a sun hat taking a pleasant stroll in Antarctica during a beautiful sunset
19	834	A push-in through an open window, entering a cozy room lit by the warm glow of a fireplace.
20	901	a toy robot wearing blue jeans and a white t shirt taking a pleasant stroll in Johannesburg South Africa during a winter storm
21	875	Macro shot. Man in an antique scuba helmet with dark glass walking out of a flower
22	991	an old man wearing purple overalls and cowboy boots taking a pleasant stroll in Johannesburg South Africa during a winter storm
23	738	A single ice cube placed in a warm drink, slowly melting and sending gentle ripples through the liquid as it transforms.
24	407	A surreal collage of a whirlwind of colorful fabrics and clothing items, fluttering and swirling in mid-air. The scene is dynamic and fashionable, with vibrant textile patterns. A sense of motion and style create a visually striking and complex scene. Pitch black background.
25	883	A mother dog gently picks up a piece of meat and carefully places it in her puppy's bowl, her eyes filled with warmth and care as she watches her little one eat.
26	14	A beautiful homemade video showing the people of Lagos, Nigeria in the year 2056. Shot with a mobile phone camera.
27	785	A pan left across a serene beach at sunrise, moving from the darkened shore to the brightening horizon.
28	231	The person's nose wrinkled in distaste as he smelled something unpleasant.
29	496	A quaint village nestled in a valley is surrounded by blooming cherry blossoms, with petals drifting through the air as villagers go about their daily activities, adding life to the scene.
30	424	Close-up of a man eating an apple.

Continued on next page

Table 22 – continued from previous page

ID	Line	Prompt
31	701	A rubber band being stretched to its maximum length and then released, snapping back to its original shape.
32	403	Cinematic trailer for a group of adventurous puppies exploring ruins in the sky
33	908	a toy robot wearing purple overalls and cowboy boots taking a pleasant stroll in Johannesburg South Africa during a beautiful sunset
34	828	A zoom-in on a drop of morning dew on a leaf, showing the reflection of the surrounding world within it.
35	927	a woman wearing blue jeans and a white t shirt taking a pleasant stroll in Johannesburg South Africa during a colorful festival
36	933	a woman wearing purple overalls and cowboy boots taking a pleasant stroll in Mumbai India during a colorful festival
37	177	A softball player sliding safely into second base.
38	35	Five gray wolf pups frolicking and chasing each other around a remote gravel road, surrounded by grass. The pups run and leap, chasing each other, and nipping at each other, playing.
39	301	A person is rapidly typing on a keyboard
40	313	A woman is search her bag trying to find something.
41	398	a man BASE jumping over tropical hawaii waters. His pet macaw flies alongside him
42	728	A crystal ball sitting on a table, with sunlight streaming through it and casting a circle of rainbow colors on the floor.
43	892	A girl is peeling an orange
44	173	a real girl franatically running a dense forest with bushes, trees, in rainy day, the animals are running after her and she is screaming and shouting
45	843	A pull-out from a child's hands holding a small seashell, moving back to show the beach and the waves around them.
46	590	A panda in a scientist's lab coat, conducting experiments with beakers and test tubes.
47	831	A push-in through a garden archway, revealing a secret, tranquil garden filled with blooming flowers.
48	631	A person holding an umbrella that turns rain into colorful confetti.
49	622	A person playing a harp strung with rainbows, creating music that colors the air.
50	260	In a vibrant theater, a magician in dazzling attire stands center stage, pulling a comically oversized rubber chicken from an ornate, old-fashioned box. His costume shimmers under the stage lights, adding to the spectacle. The crowd erupts in laughter and applause, their faces filled with joy and amazement. The magician's expression hints at mischievous delight as he holds up the rubber chicken, his performance bringing cheer to the audience.
51	296	A man is putting a ring on a woman's finger
52	783	An aerial shot of a field of blooming wildflowers, creating a patchwork of colors in the landscape.
53	750	An arc shot around a couple standing under a cherry blossom tree, petals falling around them as they embrace.
54	673	A cyclist accelerating out of the saddle during a steep climb.
55	968	an adorable kangaroo wearing a green dress and a sun hat taking a pleasant stroll in Mumbai India during a beautiful sunset
56	460	A video of a diver creating bubbles underwater, with bubbles rising and interacting with each other.
57	151	A chimp on the sidewalk doing a backflip on a skateboard.
58	752	An arc shot around a lone tree in a vast, foggy field at dawn, revealing the changing light and shadows.
59	279	A bunny puts the bright moon on its back and flies into the distance.
60	409	A dynamic motion shot of a broom morphing surreal and magically into a peacock. The handle shortens and curves into a slender neck, the bristles fanning out into a magnificent tail. The camera moves around as vibrant colors and eye-shaped patterns emerge on the expanding feathers. A small head forms at the top, complete with a delicate crest. The transformation completes as the peacock proudly displays its newly formed plumage.
61	134	a cactus with googly eyes dancing in the breeze
62	306	A pair of hands skillfully knitting a colorful scarf, the yarn winding through their fingers with each stitch.
63	283	A car crashes into a barrier at high speed.
64	323	A man is sipping a hot cup of coffee, steam rising from the mug.
65	91	A cinematic wide portrait of a man with his face lit by the glow of a TV.
66	777	An over-the-shoulder shot of a photographer adjusting their camera, framing a beautiful sunset scene.
67	374	A man is running in the forest and transforms into a wolf.
68	652	A person on a hoverboard colliding with a wall, the board stopping abruptly.
69	861	Camera tracking shot. An early 19th century scuba diver with a huge iron helmet and an iron body suit lounges on an antique lawn chair. The light is diffused and gray, casting soft shadows along the scene. The diver brings a martini glass to his helmet and puts it back down. The year is 1912. The diver is in a grassy tree-filled park. People in period-accurate dress mill around, wearing long dresses and suits, holding parasols. The diver's suit is burnished and old, held together with rusted bolts. The diver tips the martini toward his helmet and clinks the glass against the glass. The scene should feel serene and beautiful, evoking the feeling of an impressionist painting.

Continued on next page

Table 22 – continued from previous page

ID	Line	Prompt
70	114	Aerial shot of the ocean. a maelstrom forms in the water swirling around until it reveals the fiery depths below.
71	604	A person playing chess with a robot on a floating platform above the ocean.
72	910	a toy robot wearing purple overalls and cowboy boots taking a pleasant stroll in Johannesburg South Africa during a winter storm
73	790	A truck left through a bustling city street, following the flow of traffic and pedestrians during rush hour.
74	965	an adorable kangaroo wearing purple overalls and cowboy boots taking a pleasant stroll in Antarctica during a beautiful sunset
75	42	This close-up shot of a chameleon showcases its striking color changing capabilities. The background is blurred, drawing attention to the animal's striking appearance.
76	299	A woman is ascending to the sky from the ground
77	369	A paraglider descending to a landing zone.
78	137	A police helicopter hovers above a high-speed chase, guiding officers on the ground to apprehend a suspect.
79	385	a giant duck walks through the streets in Boston
80	832	A push-in towards a lone figure standing at the edge of a cliff, overlooking a vast, fog-covered valley.
81	146	A cook flipping pancakes on a griddle.
82	139	a woman jumps over a bear
83	784	An aerial view of a snow-covered mountain range, with the peaks and valleys forming intricate patterns in the snow.
84	978	an old man wearing blue jeans and a white t shirt taking a pleasant stroll in Mumbai India during a colorful festival
85	297	A man is playing the drums under the water
86	308	A person using a screwdriver to assemble a piece of furniture, carefully tightening each screw.
87	220	The person's lips curled up in a sly grin as he shared a secret joke.
88	510	A bustling ancient marketplace with merchants selling spices and fabrics.
89	408	A dynamic motion shot of a lamp transforming into a flamingo. The curved neck of the lamp elongates, its shade flattening into a delicate head. The camera circles as the base splits into two spindly legs, the bulb socket becoming a beak. Pink hues wash over the metal surface, transforming into soft feathers. The power cord coils and disappears as the transformation completes, revealing a graceful flamingo balancing on one leg.
90	207	A man is eating spaghetti
91	821	A pedestal down from the top of a waterfall, descending to show the pool of water and mist at its base.
92	104	A slow cinematic push in on an ostrich standing in a 1980s kitchen.
93	77	A FPV shot zooming through a tunnel into a vibrant underwater space.
94	566	A fish driving a tiny submarine, exploring an underwater city.
95	443	A woman drinking water from a glass, her lips touching the rim.
96	107	A cyclone of broken glass in an urban alleyway. dynamic movement.
97	325	A singer is belting out a high note on stage.
98	81	Zooming in hyper-fast to a dandelion to reveal macro dream-like abstract world.
99	737	The slow crystallization of a water droplet turning into ice on a frosty morning, with delicate patterns forming across its surface.
100	155	A chef drizzling sauce onto a plate with precision.
101	856	A little boy is riding a dragon in the sky to a castle
102	400	a dark neon rainforest aglow with fantastical fauna and animals
103	452	A young boy is playing a harmonica at sunset, with his dog sitting quietly beside him, listening.
104	286	The tall skyscapers in Hong Kong suddenly transform into a moving Gundam robot, cinematic CGI
105	969	an adorable kangaroo wearing a green dress and a sun hat taking a pleasant stroll in Mumbai India during a colorful festival
106	333	A man sighed in relief as the doctor delivered the good news.
107	390,695	pov footage of an ant navigating the inside of an ant nest
108	836	A zoom-out from a detailed shot of an intricate snowflake, pulling back to show a snowy landscape.
109	228	The person's mouth twisted in disgust as he tasted something bitter.
110	232	The person's forehead furrowed in concern as he listened to a friend's problems.
111	803	A truck right through a tranquil garden, moving past blooming flowers, trees, and a small fountain.
112	420	A marble goes through a glass cup, breaking it into pieces.
113	937	a woman wearing purple overalls and cowboy boots taking a pleasant stroll in Johannesburg South Africa during a winter storm
114	51	A white cat driving in a car through a busy downtown street with tall buildings and pedestrians in the background
115	335	A man looked away in shame when confronted with his wrongdoing.
116	499	Robots move efficiently through a futuristic laboratory, adjusting holographic displays and conducting experiments, while scientists observe and interact with the high-tech equipment.
117	357	A high-speed video of a soap bubble popping, with the soapy liquid dispersing in all directions.

Continued on next page

Table 22 – continued from previous page

ID	Line	Prompt
118	918	a toy robot wearing a green dress and a sun hat taking a pleasant stroll in Johannesburg South Africa during a colorful festival
119	699	A magician takes off his performing mask.
120	271	Penguins roller skating
121	879	An astronaut fighting a monster
122	493	In an ancient library, books float and glow as they drift through the air, occasionally landing softly on the tables, where curious individuals reach out to read their contents.
123	164	A chef skillfully tossing a salad in a bowl.
124	891	A mother is peeling an apple for her daughter
125	106	Zooming through a dark forest with neon light flora lighting up.
126	565	A horse wearing roller skates, gracefully gliding through a city park.
127	215	The person walked into the room, his face lighting up with a warm smile.
128	193	A pair of lovebirds preening each other's feathers.
129	634	A person skydiving from a hot air balloon into a sea of clouds.
130	46	Tiltshift of a construction site filled with workers, equipment, and heavy machinery.
131	444	A woman singing softly to a baby, her lips forming gentle words.
132	156	A frog that gets kissed and turns into a chocolate milkshake.
133	581	A kangaroo wearing boxing gloves, sparring with a punching bag in a gym.
134	131	Hyperrealistic monster that closes its mouth
135	458	A time-lapse video of a river flowing through a forest, with changing water levels and currents.
136	797	A pan right across a field of tall grass swaying gently in the wind, with a setting sun in the background.
137	725	Sunlight piercing through a canopy of bamboo, casting long, linear shadows and patches of light on the forest floor.
138	574	A peacock wearing a crown, sitting on a throne and holding court with other animals.
139	90	A FPV drone shot through a castle on a cliff.
140	38	A grandmother with neatly combed grey hair stands behind a colorful birthday cake with numerous candles at a wood dining room table, expression is one of pure joy and happiness, with a happy glow in her eye. She leans forward and blows out the candles with a gentle puff, the cake has pink frosting and sprinkles and the candles cease to flicker, the grandmother wears a light blue blouse adorned with floral patterns, several happy friends and family sitting at the table can be seen celebrating, out of focus. The scene is beautifully captured, cinematic, showing a 3/4 view of the grandmother and the dining room. Warm color tones and soft lighting enhance the mood.
141	774	A close-up of a key turning in a lock, showing the subtle movements of the key and the intricate details of the mechanism as it turns into place.
142	23	A cat waking up its sleeping owner demanding breakfast. The owner tries to ignore the cat, but the cat tries new tactics and finally the owner pulls out a secret stash of treats from under the pillow to hold the cat off a little longer.
143	782	An aerial perspective of a group of dolphins swimming near the surface of a crystal-clear ocean, their movements synchronized.
144	596	A person walking up a staircase made of clouds leading to a floating castle.
145	322	A little child let out a big yawn
146	863	Over the shoulder camera shot. A huge lizard creature sits in a midcentury orange swivel chair. The light is dim and volumetric, casting an eerie glow across the scene. The creature uses its arms to maniacally push buttons on a gigantic control panel. Above the control panel is a panoramic window looking out and down on 1940s new york city. The room should invoke midcentury science fiction aesthetics, like rusty orange colors, bright flashing control buttons, and space-age flair. As the creature continues to quickly push buttons, the New York City scene out of the window moves closer, as though the creature is in a gigantic robot stomping through the city. The scene should give the feeling of frantic action, highlighting the intensity of piloting a giant robot. The scene should take inspiration from midcentury japanese monster films.
147	281	Whilst a man and woman are walking through a city street in a dream, the man shows the woman how to fold the entire street upwards at a 90-degree angle and connect it with the sky. This creates a visually stunning effect, with the buildings and road bending and defying gravity. The scene highlights the limitless possibilities and creativity within the dream world.
148	764	A first-person view of an artist sketching in a notebook, the pencil moving swiftly across the page as the drawing takes shape.
149	393	an alien blending in naturally with new york city, paranoia thriller style, 35mm film
150	687	A woman is playing a soft melody on his lute while sitting by the fountain in the castle courtyard.
151	830	A push-in through a dense crowd at a festival, moving towards a performer on stage who is captivating the audience.
152	82	Internal window of a train moving at hyper-speed in an old European city.
153	32	New York City submerged like Atlantis. Fish, whales, sea turtles and sharks swim through the streets of New York.
154	199	A martial artist breaking a board with a powerful punch.

Continued on next page

Table 22 – continued from previous page

ID	Line	Prompt
155	153	A fish walking into a coffee shop and asking for a cup of coffee.
156	872	A computer mouse with legs running on a treadmill
157	206	a woman eating ice scream
158	63	Carefully pouring the milk into the cup, the milk flowed smoothly and the cup was gradually filled with a milky white color
159	184	A soccer goalie making a diving save with outstretched arms
160	171	A penguin flies into the mouth of a blue whale breaking the surface of the water.
161	471	A close-up of a drink being poured over ice, capturing the detailed flow and interaction with the ice cubes.
162	128	A curious cat peering out from a cozy hiding spot.
163	975	an adorable kangaroo wearing a green dress and a sun hat taking a pleasant stroll in Antarctica during a colorful festival
164	257	A bird made of fresh oranges rushes out of the orange
165	570	A bear wearing a superhero cape, flying through the sky over a bustling city.
166	718	Light filtering through a canopy of autumn leaves, casting warm, dappled patterns of yellow, orange, and red onto the ground.
167	654	A person on a scooter colliding with a park bench, the scooter tipping over.
168	677	A bobsled team accelerating down an icy track.
169	816	A tilt-down from the towering peak of a mountain to the winding path leading up to it.
170	255	Camera tracking shot. A man walking down a city street, holding a coffee cup in his hand. He is wearing a dark suit and red tie.
171	379	A rose is growing out of a stone
172	886	A mother bird feeding her chicks in the nest, delicately placing food into their wide-open beaks as they chirp eagerly.
173	217	The person's eyebrows furrowed in concentration as he worked on a puzzle.
174	218	The person's mouth dropped open in surprise as he watched a magic trick.
175	929	a woman wearing blue jeans and a white t shirt taking a pleasant stroll in Antarctica during a beautiful sunset
176	578	A fox wearing a pirate hat and eyepatch, steering a ship through a stormy sea.
177	823	A pedestal down through a field of sunflowers, showing their tall stalks and bright petals against the sky.
178	337	A man grinned with satisfaction after completing the challenging task.
179	747	A thin sheet of ice on a lake cracking and breaking as the sun warms it, creating a mosaic of shifting patterns.
180	122	A bird landing on water and turning into a fish.
181	384	a photorealistic video of a butterfly that can swim navigating underwater through a beautiful coral reef
182	266	A red panda taking a bite of a pizza
183	202	A child's face lighting up with joy as they blow out the candles on their birthday cake.
184	191	A person opens a book and turns it upside-down and characters from the book begin to fall out of it.
185	244	The person jumps up and down excitedly, expressing happiness through dance moves.
186	208	A person takes a big bite of a juicy burger, the meat and cheese filling his mouth.
187	935	a woman wearing purple overalls and cowboy boots taking a pleasant stroll in Johannesburg South Africa during a beautiful sunset
188	431	A man pushes away a huge stone with superhuman strength.
189	109	Highly detailed close up of a bacteria.
190	573	A goat operating a food truck, serving gourmet grilled cheese sandwiches to a line of animals.
191	472	A mesmerizing video of a whirlpool forming in a sink as water drains.
192	97	An astronaut walking between stone buildings.
193	610	A person doing yoga on top of a giant lily pad in the middle of a serene pond.
194	636	A person riding a giant tortoise through a desert of glass sand.
195	614	A person playing hopscotch on the rings of Saturn.
196	938	a woman wearing purple overalls and cowboy boots taking a pleasant stroll in Antarctica during a beautiful sunset
197	844	A pull-out from a dancer's feet moving gracefully, expanding to show the entire stage and audience.
198	463	A close-up of paint being mixed, showing the detailed interaction of colors and textures.
199	442	A person chewing on a pencil while deep in thought.

I Broader Societal Impact

VAR-Q reduces the KV-cache memory footprint of visual autoregressive models, enabling some inference configurations that otherwise exceed device memory. These savings may broaden access for researchers and creators with limited GPU memory. They do not by themselves establish lower inference time, energy consumption, or hardware cost; quality and runtime depend on the model and precision.

Broader access to visual generation also carries risks, including deceptive synthetic media, non-consensual imagery, and the propagation of biases inherited from the underlying models. KV-cache quantization does not address these risks. Deployment should retain the base model's safeguards and incorporate appropriate content provenance, consent protections, and abuse monitoring.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [No]

   Justification: The structural findings and quantization method are supported by Sections 3.1–3.4 and 4. However, the maximum batch-size claim is not fully reconciled with the configuration-dependent deployment measurements in Appendix D.1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Appendix B discusses limitations of video metrics under outlier ablation, and Appendix A states the conditions and scope of the RoPE bounds. Appendix D.1 distinguishes memory-capacity gains from runtime overhead.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Appendix A defines the token-pair amplitudes and shared-angle setting and provides the lower- and upper-bound proofs for Eq. (4).

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No]

   Justification: Sections 3.4 and 4, Appendix F, and the public code describe the quantization method and evaluation setup. However, the configuration differences among the batch-capacity and joint-quantization measurements are not fully resolved, so the paper does not yet specify a complete reproduction procedure for every main deployment claim.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code, installation instructions, inference configurations, and VAR/Infinity evaluation scripts are publicly available in the `VAR-Q` repository; third-party models and datasets must be obtained from their upstream sources. The release does not provide a complete end-to-end reproduction workflow for all reported video evaluations and deployment comparisons.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sections 3.4 and 4 describe grouping, the default sub-block ratio, backbones, benchmarks, metrics, and evaluation hardware. Appendix D specifies workloads and timing procedures; the method requires no retraining or calibration.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The reported comparisons do not include error bars, confidence intervals, or statistical significance tests supporting the main experimental claims.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Section 4 identifies NVIDIA A100 80 GB GPUs, and Appendix D.1 reports generation latency and the timing protocol. The paper does not report complete GPU-hours for all experiments or an estimate of total project compute, including preliminary and failed runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research utilizes standard open-source datasets and models and follows established responsible AI practices.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix I discusses potential benefits of reduced memory requirements and risks involving deceptive media, non-consensual imagery, and inherited model biases.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [No]

    Justification: Appendix I recommends retaining base-model safeguards and using provenance and abuse-monitoring measures. The paper does not document additional release-specific safeguards implemented or evaluated for `VAR-Q`; VLM-based quality evaluation is not a misuse safeguard.

    Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

    Justification: The paper cites the models and benchmarks used, and the public repository distinguishes its MIT-licensed code from third-party assets governed by their own terms. The paper does not enumerate the exact license and version for every external model, dataset, and code dependency.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new software is documented in the `VAR-Q` repository, including installation, supported backends, configuration fields, inference examples, evaluation scripts, and an MIT license. Third-party model checkpoints and datasets are obtained separately; this work does not introduce a new dataset.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper uses a "VLM-as-judge" approach (Claude Opus 4.7) for video evaluation instead of human crowdsourcing.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No human subjects were involved in the research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Section 4 identifies Claude Opus 4.7 as the VLM judge for video evaluation. Appendix E provides the frame-sampling procedure, scoring instructions, and prompt template.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.