POSITIONAL SPECIALIZATION FOR AUTOREGRESSIVE VIDEO KV-CACHE COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

KV-cache compression for autoregressive video is usually framed as deciding how much of the past to preserve. Using persistent, position-matched interventions, we show instead that different temporal positions require different kinds of information. Distant states, including pinned and mid-lag positions, can be spatially collapsed from 1,560 tokens to a single token with negligible loss, whereas the same collapse on recent states causes severe frame-to-frame flicker. Spatial compressibility, however, does not imply dispensability: evicting distant states degrades image quality and nearly halves motion, whereas retaining each as a single token does not, and replacing them with another video's cache shifts output semantics increasingly with semantic distance. These findings motivate **AxisKV**, a two-axis allocation that spatially compresses distant states while quantizing recent states. At $8\times$ cache compression on Self-Forcing, **AxisKV** outperforms matched-memory uniform pooling and, against uniform quantization, trades a loss in aesthetic quality for higher image quality; it also remains usable at $12\times$, where uniform quantization has no valid operating point. The positional specialization transfers across related autoregressive video generators.

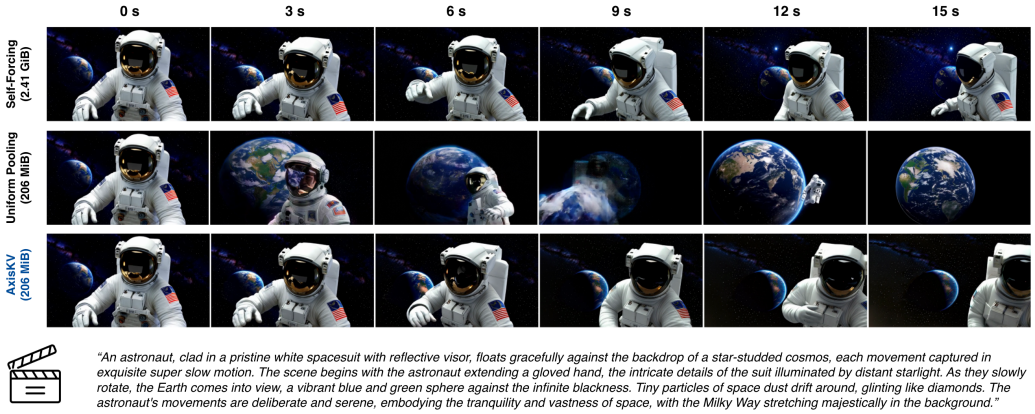


Figure 1: **Position-aware cache compression preserves long-horizon generation under a tight memory budget.** At the same 206 MiB cache size, uniform spatial pooling causes substantial semantic and compositional drift over time, while **AxisKV** remains close to the uncompressed Self-Forcing reference throughout the 15 s rollout.

1 INTRODUCTION

Autoregressive (AR) video diffusion models have made long-horizon generation increasingly practical. Methods such as CausVid Yin et al. (2025), Self-Forcing Huang et al. (2026), Rolling Forcing Liu et al. (2026), and LongLive Yang et al. (2026a) generate video incrementally while reusing key-value (KV) states from previous frames or chunks. This persistent cache is also a major memory bottleneck: for Wan2.1-T2V-1.3B at 480×832 resolution, one cached latent frame occupies roughly 290 MB across 30 layers. This has motivated cache compression through token pruning,

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

eviction, and merging (Li et al., 2026b; Chen et al., 2026a; Ji et al., 2026), low-bit quantization (Xi et al., 2026; Tuncer et al., 2026), low-rank representations (Yesiltepe et al., 2026a), and position- or head-dependent allocation (Mao et al., 2026; Zhao et al., 2026b; Chen et al., 2026b).

These methods increasingly treat historical states as heterogeneous. Retention policies preserve early frames as attention sinks alongside recent context, while newer methods vary allocation across temporal regions or heads (Mao et al., 2026; Zhao et al., 2026b; Chen et al., 2026b; Ji et al., 2026). Early states have also been described as a global context anchor (Liu et al., 2026), a reference that can lock scene composition (Dalva & Yanardag, 2026), or an anchor for temporal coherence (Yi et al., 2025). Yet these roles are inferred mainly from attention, redundancy, architectural heuristics, or end-to-end quality. What remains unclear is not simply *which positions matter*, but *what information each position must preserve*, and whether that requirement changes with temporal distance. To our knowledge, prior work has not causally isolated these requirements under position-matched interventions.

Probing these requirements in an autoregressive generator is nontrivial. Replacing activations with noise or zeros can introduce distribution shift unrelated to the information being tested (Zhang & Nanda, 2024). Moreover, a cache perturbation changes subsequent frames, which are written back into the cache, so a one-step intervention misses its effect on the evolving history. We therefore use persistent, position-matched interventions, comparing replacements against the recipient's own states processed identically, to probe spatial detail, numerical precision, content identity, and slot presence.

On Self-Forcing (Huang et al., 2026), the probes reveal a sharp functional dissociation across temporal positions (Figure 2). Pinned and mid-lag states can be collapsed from 1,560 tokens to one with almost no loss in quality or stability, whereas collapsing recent states severely disrupts temporal continuity. In contrast, two-bit quantization is inexpensive at all positions, revealing a clear asymmetry between spatial detail and numerical precision. Yet distant states are not disposable. Removing pinned slots partially freezes the rollout, whereas removing the same number of mid-lag slots does not. Their failure modes also differ: perturbing recent states breaks temporal continuity, while perturbing distant states shifts semantics with little flicker; donor caches at distant positions induce larger semantic shifts as donor content becomes less similar. Cache positions therefore differ not merely in importance, but in *what information they preserve* and *which output property they control*.

These findings suggest a different compression principle: compress different temporal positions along different axes, not merely at different rates. We spatially compress distant states while preserving recent spatial structure and reducing its numerical precision instead. The allocation is deliberately simple; the contribution is the principle, not a new pooling operator or quantizer. At $8\times$ and $12\times$ compression on Self-Forcing (Huang et al., 2026), it changes no VBench-Long (Huang et al., 2025) dimension by more than 1.4 points relative to the uncompressed cache and outperforms uniform pooling at matched memory in quality and stability. At $12\times$, matching the same budget with uniform quantization would require 1.33 bits per value. The analysis also exposes the method's trade-off: spatially collapsing pinned states costs aesthetic quality at every budget, so the allocation is not uniformly better than low-bit quantization. Indeed, when three-bit uniform quantization fits the budget, it is preferable. Across three additional generators, the recent-state spatial requirement and the stability advantage over pooling transfer, while the cost of compressing distant states remains model-dependent and is consistent with the corresponding probe.

Our contributions are threefold: (1) a causal probing protocol for recurrent generative caches that captures autoregressive compounding while controlling replacement source; (2) a position-dependent functional organization of the cache, with asymmetric spatial and precision requirements and distinct temporal and semantic failure modes; and (3) a cache allocation derived from this organization that chooses *which axis* to compress at each position, rather than only *how aggressively*.

2 Related Work

Autoregressive long-video generation. Autoregressive video diffusion models generate frames or chunks conditioned on previously generated states, enabling long-horizon and streaming synthesis (Yin et al., 2025; Huang et al., 2026; Liu et al., 2026; Yang et al., 2026a; Li et al., 2026c). Be-

cause errors accumulate and historical context becomes increasingly difficult to maintain over long rollouts, subsequent methods have introduced increasingly structured mechanisms for maintaining historical information. These include persistent or enlarged attention sinks (Liu et al., 2026; Yang et al., 2026a; Yi et al., 2025), multi-timescale memories (Chen et al., 2026c), evolving or recurrent memory states (Kim et al., 2026; Dalva & Yanardag, 2026), anchor-based re-caching (Yang et al., 2026b), and training objectives that directly supervise how historical context is encoded (Zhuang et al., 2026), and adaptive memory updates that continually refresh persistent context (Lu et al., 2026). Other approaches modify positional encoding, cache indexing, or rollout dynamics to stabilize arbitrarily long generation (Yesiltepe et al., 2026b; Li et al., 2026a; Wu et al., 2025). These methods increasingly assign distinct roles to different parts of history, such as global anchors for early states and local temporal context for recent ones. However, such functional interpretations are typically inferred from attention statistics, architectural design choices, or end-to-end rollout quality. We instead intervene directly on cached states to identify what information they causally provide and how that requirement varies with temporal position.

KV-cache compression for autoregressive video. The growing memory cost of historical KV states has motivated compression along several complementary axes. Token-selection methods prune, evict, or merge historical states using attention, salience, redundancy, or estimates of future utility (Li et al., 2026b; Chen et al., 2026a; Yi et al., 2025; Luo et al., 2026; Zhao et al., 2026a; Samuel et al.; Ma et al., 2026). A parallel line exploits heterogeneous attention behavior across heads, assigning head-specific cache budgets or retention rules (Guo et al., 2026; Ji et al., 2026; Tian et al., 2026; Chen et al., 2026b; Xu et al., 2026; Lv et al., 2026). Other methods reduce the representation cost of retained states through low-bit quantization (Xi et al., 2026; Tuncer et al., 2026), low-rank latent representations (Yesiltepe et al., 2026a), parametric absorption of context (Fu et al., 2026), or compact recurrent summaries (Kim et al., 2026; Yi et al., 2026). Recent methods also employ position-dependent cache policies: PackForcing aggressively compresses intermediate history while preserving sink and recent states (Mao et al., 2026), while Relax Forcing and Pyramid Forcing similarly structure allocation across temporal regions (Zhao et al., 2026b; Chen et al., 2026b). These approaches demonstrate that non-uniform cache allocation can be effective, but their policies are motivated mainly by attention patterns, redundancy, salience, or end-to-end generation behavior. We ask a different question: *what information must each cache position preserve?* By causally probing spatial detail, numerical precision, content identity, and slot presence, we derive an allocation that varies the *compression axis*, rather than only the compression rate, across temporal positions.

Causal and activation interventions. Causal mediation analysis, interchange interventions, causal tracing, activation patching, and path patching have been used to localize model computations and test the functional role of internal representations (Vig et al., 2020; Geiger et al., 2022; Meng et al., 2022; Goldowsky-Dill et al., 2023; Geiger et al., 2025). At the same time, several studies have shown that intervention-based conclusions can depend strongly on the corruption distribution, replacement value, evaluation metric, and intervention design (Zhang & Nanda, 2024; Heimersheim & Nanda, 2024; Li & Janson, 2024). In particular, artificial noise, zero ablation, or other off-distribution replacements may remove the targeted information while simultaneously introducing unintended signals, confounding causal attribution. Intervention-based analyses have also begun to probe diffusion models, including cross-attention and semantic grounding (Maliha & Hougen, 2026). Autoregressive generation introduces an additional challenge: a perturbed state affects future outputs, which are then written back into the context. We therefore use position-matched controls and persistent interventions to measure compounded causal effects over the rollout rather than only immediate one-step responses.

3 Causal Analysis and Position-Dependent Cache Allocation

3.1 Persistent, Position-Matched Probing

An AR video diffusion model generates a latent video $x_{1:N}$ chunk by chunk. At each denoising step, every attention layer reads a KV cache containing keys and values from earlier latent frames ($1{,}560$ tokens per frame for Wan2.1-T2V-1.3B at 480×832). Following common training-free cache designs (Xiao et al., 2024; Ji et al., 2026), the first latent frames remain pinned throughout the rollout, while recent history is maintained in a sliding window. Let τ_k denote the index of the first latent

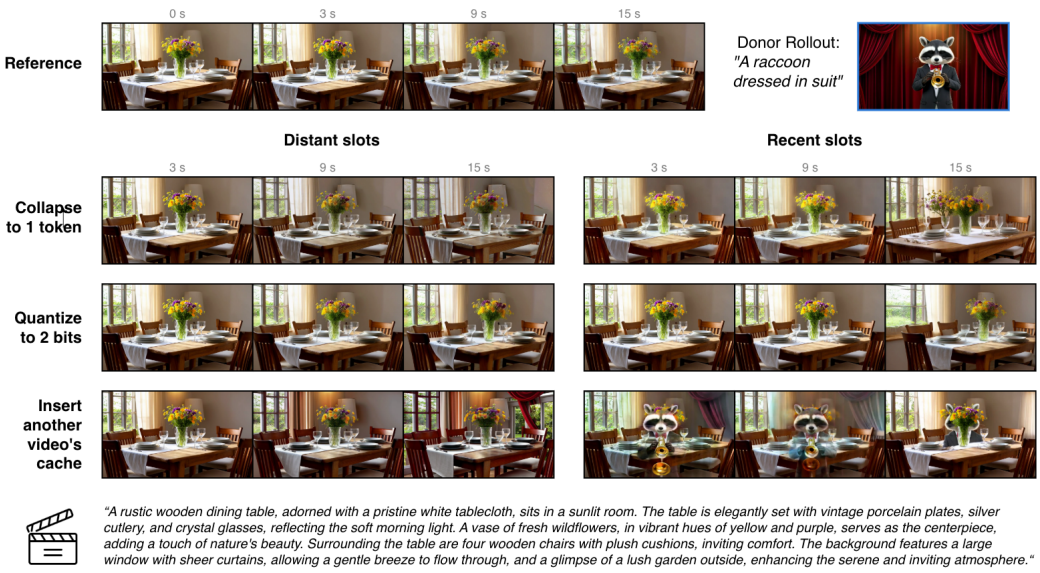


Figure 2: **Cache positions require different kinds of information.** We apply the same interventions to distant (pinned and mid-lag) or recent cache slots in a representative 15 s rollout. Collapsing distant slots to one token largely preserves the scene, whereas collapsing recent slots disrupts temporal continuity; two-bit quantization is tolerated at both positions. Transplanting another video's cache at distant slots lets donor content surface gradually, whereas the same transplant at recent slots quickly redirects generation toward the donor.

frame in the chunk generated at step k, and define the lag of cached frame i as $\ell_k(i) = \tau_k - i$. The nine history slots then split into

$$G_{\text{pin}} = \{1, 2, 3\}, \qquad G_{\text{rec}}(k) = \{i : 1 \le \ell_k(i) \le 3\}, \qquad G_{\text{mid}}(k) = \{i : 4 \le \ell_k(i) \le 6\}, \quad (1)$$

so that pinned slots are fixed in absolute position while the other two groups track fixed lags as the window advances.

We intervene only on the cache presented to attention. Let C_k be the stored cache at step k and $\widetilde{C}_k$ the cache read by the attention layers. Abstracting one autoregressive step,

$$x^{(k)} = F_\theta(c, \epsilon_k, \widetilde{C}_k), \qquad C_{k+1} = U(C_k, x^{(k)}), \quad (2)$$

where F_θ generates the next latent chunk and U writes its states into the cache. A perturbation at step k therefore affects not only the current chunk but the future history generated from it. For an intervention beginning at step k_0,

$$\widetilde{C}_k = \begin{cases} \mathcal{I}_{T,G(k)}(C_k), & k \in \mathcal{K}, \\ C_k, & \text{otherwise}, \end{cases} \qquad \mathcal{K} = \begin{cases} \{k_0\}, & \text{one-step}, \\ \{k : k \ge k_0\}, & \text{persistent}, \end{cases} \quad (3)$$

where $\mathcal{I}_{T,G(k)}$ applies operator T to the targeted group in the copy passed to attention, at every layer and denoising step, while leaving C_k unchanged. One-step interventions are common in activation patching (Zhang & Nanda, 2024; Heimersheim & Nanda, 2024), but measure only the immediate response. We instead use persistent interventions because the consequences of a cache perturbation are themselves written into future history. We set k_0 to the first step at which all nine history slots are occupied and the three groups in Eq. 1 are disjoint.

Because recent and mid-lag slots advance with the window, an intervention is either *live*, recomputed from the current slot at every read, or *frozen*, reusing a snapshot from a fixed step; the two coincide for pinned slots. Compression operators—spatial pooling and uniform affine quantization with the analytic softmax correction of Tuncer et al. (2026)—are applied live, matching a compressor that stores $T(\cdot)$ on write. Eviction differs because it changes the cache configuration itself and therefore acts from the first chunk.

Table 1: **The interventions scored on the benchmark** ($n = 100$, 15 seconds). Colours mark the cache position. Three conventions hold for every table in the paper: entries are the paired change from the unmodified rollout, and where a table has an uncompressed first row that row is absolute and the rest are changes from it; grey marks a difference that is not significant at $p < 0.05$ in a two-sided paired t-test; chunk discontinuity is a single mean per arm and carries no such test.

operator	group	subject ↑	backgr. ↑	smooth. ↑	dynamic	aesth. ↑	imaging ↑	disc. ↓	semantic ↑
collapse to 1 token	**pinned**	+0.04	-0.09	-0.01	-0.50	-0.39	+0.27	-0.046	-0.026
	mid-lag	-0.06	-0.01	-0.01	-1.62	+0.18	+0.06	-0.043	+0.004
	recent	-2.45	-1.26	-1.48	+10.75	-2.22	-2.90	+1.610	-0.003
quantize to 2 bits	**pinned**	-0.06	-0.03	-0.00	-3.00	-0.30	-0.48	-0.013	+0.001
	mid-lag	-0.06	-0.04	-0.02	+0.25	-0.02	-0.14	+0.002	-0.003
	recent	-0.32	-0.23	-0.14	+2.00	-0.31	-0.54	+0.153	-0.007
evict	**pinned**	-0.03	-0.25	-0.17	-7.38	-0.91	-0.49	+0.404	-0.032
	mid-lag	-0.07	-0.05	-0.01	+1.50	+0.12	-0.25	-0.112	+0.002

To test whether a position requires the *right* content, we transplant keys and values from position-matched slots of a real donor rollout rather than using off-distribution zeros or noise (Zhang & Nanda, 2024; Heimersheim & Nanda, 2024; Li & Janson, 2024). Donor states are necessarily frozen, so we compare them against the recipient's own states frozen at the same positions and step; the two conditions therefore differ only in content identity. We report the paired effect

$$\Delta_T(G) = \mathbb{E}_c \left[m\left(x_{1:N}^{T,G}\right) - m(x_{1:N}) \right] \tag{4}$$

over 100 VBench prompts (Huang et al., 2024) in the extended form released with CausVid (Yin et al., 2025), with initial noise fixed per prompt and rollouts of about 15 s. Appendix B gives the remaining details.

3.2 What Each Position Requires

We probe four distinct requirements of a cached state: spatial detail, numerical precision, content identity, and presence. We remove spatial detail by collapsing all 1,560 tokens of a slot into a single summary token, reduce precision through two-bit quantization, replace content identity by transplanting another rollout's states into position-matched slots, and remove the slot entirely through eviction. Table 1 scores the three compression-like operators on the six VBench-Long dimensions, chunk discontinuity and semantic consistency. The transplant is a counterfactual probe rather than a compression scheme, so we report it qualitatively: Figure 2 shows it alongside representative rollouts for the other three.

The strongest asymmetry is spatial. Collapsing pinned and mid-lag states to one token leaves generation largely intact: no VBench-Long dimension changes significantly at mid-lag slots, and at pinned slots the benchmark effects are small. Semantic consistency is the exception, falling by 0.026 at pinned slots. Applying the same operation to recent states instead causes substantial degradation, including -2.45 points in subject consistency, -1.48 in motion smoothness, and -2.90 in imaging quality. This sensitivity is specific to spatial structure rather than compression in general. Two-bit quantization is inexpensive throughout the cache: even at recent slots, the three consistency dimensions fall by only 0.32, 0.23, and 0.14 points. Recent states therefore do not simply require *more* cache capacity; they specifically require spatial detail. Spatial resolution and numerical precision are thus asymmetric compression axes: aggressive spatial collapse is tolerated only at distance, whereas low precision remains viable across temporal positions.

Spatial compressibility, however, does not imply dispensability. Pinned states can be reduced to a single token with little effect, yet removing them entirely reduces dynamic degree by 7.38 points and degrades background consistency, motion smoothness, and aesthetic quality, partially freezing the rollout. Evicting the same number of mid-lag states produces no significant change, ruling out reduced cache size alone as the explanation. The one-token and eviction interventions therefore separate spatial content from slot presence: pinned states require little spatial detail, but must remain represented.

3.3 Position Determines the Failure Mode

The positional differences in Table 1 are not merely differences in magnitude; they correspond to qualitatively different failures. Collapsing recent states degrades subject consistency, background consistency, motion smoothness, aesthetic quality, and imaging quality, while increasing dynamic degree by 10.75 points. The increase reflects instability rather than improved motion: chunk discontinuity rises by 1.61 over an unmodified value of 2.08, so the additional movement is concentrated at chunk boundaries rather than distributed through the rollout. Figure 2 shows this behavior directly, with the rollout remaining active but progressively losing temporal and spatial coherence. Recent states therefore appear to be critical for maintaining a coherent continuation.

Collapsing pinned or mid-lag states to one token leaves the rollout visually close to the reference, despite removing nearly all spatial structure. Their content nevertheless remains causally active: collapsing the pinned states costs 0.026 of semantic consistency while leaving chunk discontinuity no worse than the unmodified rollout, whereas collapsing recent states leaves semantic consistency untouched (-0.003, not significant) and raises chunk discontinuity by 1.61. The two positions therefore control different output properties rather than differing in importance (Figure 2).

These interventions reveal a position-dependent functional organization of the cache. Recent states are sensitive to the loss or replacement of spatially resolved content, while distant states tolerate extreme spatial compression yet remain semantically active. This distinction motivates allocating different compression axes, rather than merely different compression rates, across temporal regions.

3.4 From Analysis to Cache Allocation

The analysis above yields a simple allocation principle: compress each temporal region along the axis it tolerates. Distant states tolerate extreme spatial collapse but pinned states should remain represented, whereas recent states require spatial structure but tolerate low numerical precision. We therefore collapse each pinned and mid-lag slot to one spatial token, retain recent slots at full spatial resolution, and allocate the remaining memory budget to their precision (Figure 3). The design intentionally introduces no new pooling operator or quantizer; its contribution is the position-dependent choice of compression axis.

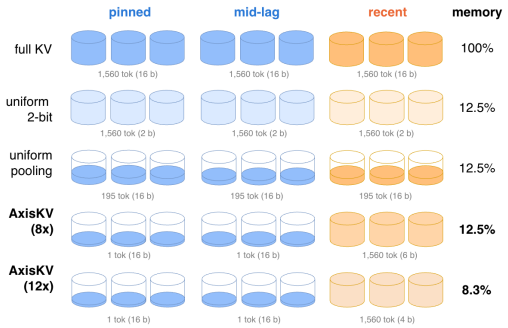


Figure 3: **AxisKV allocates different compression axes to different cache positions.** AxisKV collapses pinned and mid-lag states spatially to one token and preserves the full spatial resolution of recent states while reducing their precision.

Let $s(p)$ denote the number of spatial tokens retained at cache position p ($S = 1{,}560$ at full resolution) and $b(p)$ its precision in bits. Relative history memory is

$$\bar{M} = \frac{\sum_p s(p)b(p)}{\sum_p S \cdot 16}. \tag{5}$$

Uniform quantization fixes $s(p) = S$ and varies b, while uniform pooling fixes $b(p) = 16$ and varies s. Our allocation instead uses

$$s(p) = 1,\ b(p) = 16 \quad \text{for pinned and mid-lag states,} \qquad s(p) = S,\ b(p) = b_{\text{rec}} \quad \text{for recent states.} \tag{6}$$

Because the six distant slots together retain only six tokens, they consume negligible memory: 0.04% of the uncompressed history. The recent-state precision b_{rec} is therefore determined directly by the target budget rather than tuned independently. An $8\times$ memory reduction gives $b_{\text{rec}} = 6$, while $12\times$ gives $b_{\text{rec}} = 4$. Matching the latter budget with uniform quantization would require 1.33 bits per value, below the two-bit integer operating point considered here.

Collapsed slots store one mean key and value while retaining their cache positions; quantization is uniform affine per token, head and group of 32 channels, with the analytic softmax correction of

Table 2: **30-second generation on Self-Forcing** (30 seconds, $n = 100$). Memory is the history KV stored between steps, under one accounting for every method (Appendix K); Dummy Forcing's is a range because it splits its heads per prompt. Rows above the rule run in the authors' released implementations, rows below in ours. Shading marks the best three values in a column; dynamic degree is unshaded because higher is not better here (Section 3.3).

Method	Memory ↓	subject ↑	backgr. ↑	smooth. ↑	dynamic	aesth. ↑	imaging ↑	disc.↓
Full KV	4.82 GiB	97.43	96.52	98.74	36.00	59.78	71.05	3.496
Forcing-KV	715 MiB	97.39	96.32	98.40	47.47	60.85	71.66	2.893
Dummy Forcing	274–548 MiB	97.70	96.72	98.66	45.87	62.40	71.59	3.396
Deep Forcing	4.69 GiB	97.23	96.68	98.53	45.33	61.17	71.06	2.887
∞-RoPE	822 MiB	97.34	96.47	98.28	52.47	60.45	71.96	2.429
AxisKV ($8\times$)	308 MiB	97.64	96.60	98.79	43.93	60.00	72.31	2.743
AxisKV ($12\times$)	206 MiB	97.60	96.54	98.74	43.73	59.60	71.88	2.853

Tuncer et al. (2026). Both operators are training-free and modify only the cache. We report stored-history memory from Eq. 5, not wall-clock acceleration; implementation details and the equivalence between read- and write-time compression are given in Appendix C.4.

4 Experiments

4.1 Setup

All experiments use Self-Forcing (Huang et al., 2026), a Wan2.1-T2V-1.3B backbone (Wan et al., 2025) distilled to four DMD steps (Yin et al., 2024b;a), whose cache holds nine history slots, three pinned, three mid-lag, three recent, beside the chunk being denoised. We generate 29.8 s per prompt, six times the trained window, for the 100 VBench prompts (Huang et al., 2024) used in Section 3; every arm receives the same prompt and the same initial noise, so all differences are paired with $n = 100$. We report VBench-Long (Huang et al., 2025) on its six dimensions and Chunk Discontinuity (Ji et al., 2026), the protocol prior cache designs are evaluated on, and, where a claim concerns whether the end of a video still depicts what its beginning did, the CLIP semantic consistency, which no VBench dimension measures. Memory is the stored history KV of Eq. 5, computed from the policy rather than measured. Appendix A gives the full configuration. We compared our **AxisKV** with Full KV, Forcing-KV (Ji et al., 2026), Dummy Forcing (Guo et al., 2026), Deep Forcing (Yi et al., 2025), and ∞-RoPE (Yesiltepe et al., 2026b). Appendix K derives each method's memory under one accounting.

4.2 Results

Comparison with existing methods. Table 2 gives the benchmark. **AxisKV** stores 308 MiB of history at $8\times$ and 206 MiB at $12\times$, a sixteenth to a twenty-fourth of Full KV, and is not significantly worse than Full KV on any VBench-Long dimension while gaining imaging quality ($+1.26$, $t = 3.8$). Against its own uncompressed sliding window, the arm that isolates the cache policy from every implementation difference, it gains imaging quality ($+1.01$, $t = 3.3$), loses aesthetic quality (-0.73, $t = -2.2$) and is level on the consistency dimensions. The published designs all keep more memory: Forcing-KV $2.3\times$ our budget, ∞-RoPE $2.7\times$, Deep Forcing about what Full KV keeps, Dummy Forcing comparable but prompt-dependent; and the pattern against them is consistent: **AxisKV** is ahead on the consistency dimensions and on imaging quality (subject, smoothness and imaging against Forcing-KV, Deep Forcing and ∞-RoPE, $t = 2.2$ to 8.6) and behind on aesthetic quality against every one of them, most clearly against Dummy Forcing (-2.40, $t = -6.4$). Dynamic degree is highest for the least stable arms, as Section 3.3 predicts, so we read it next to chunk discontinuity rather than on its own.

At matched memory. Table 3 compares allocations inside our implementation, where the cache policy is the only thing that differs. At $\bar{M} = 0.125$ the allocation is statistically indistinguishable from the uncompressed cache on subject and background consistency and on motion smoothness, and it gains imaging quality ($+1.01$); the two uniform schemes at the same 308 MiB both lose all

Table 3: **Quality against memory** (30 seconds, $n = 100$). Shaded cells are the best value in a column within a matched-memory group ($\bar{M} = 0.125$ and 0.083); the rows above them differ in memory and are not a within-group comparison. Dynamic degree is unshaded because higher is not better here (Section 3.3).

$\bar{M}$	Allocation	Memory ↓	subject ↑	backgr. ↑	smooth. ↑	dynamic	aesth. ↑	imaging ↑	disc.↓
1.000	uncompressed history	2.41 GiB	97.51	96.66	98.80	45.13	60.73	71.29	2.640
0.375	uniform 6-bit	904 MiB	−0.03	−0.02	+0.02	+1.47	+0.06	+0.13	+0.004
0.334	**AxisKV**, 16-bit recent	805 MiB	+0.08	−0.06	−0.03	−0.60	−0.77	+0.68	+0.032
0.250	uniform 4-bit	617 MiB	−0.05	−0.05	−0.04	+1.73	−0.13	−0.26	+0.028
0.188	uniform 3-bit	463 MiB	+0.00	−0.01	−0.02	−1.40	−0.06	−0.22	+0.139
0.167	**AxisKV**, 8-bit recent	412 MiB	+0.14	−0.09	−0.03	−1.33	−0.83	+0.69	+0.147
0.125	uniform 2-bit	308 MiB	−0.47	−0.49	−0.14	−1.13	−1.60	−1.87	+0.226
0.125	uniform pooling, 195 tok.	308 MiB	−3.35	−1.67	−0.94	+22.27	−3.25	−2.56	+2.214
0.125	**AxisKV**, 6-bit recent	308 MiB	+0.13	−0.06	−0.01	−1.20	−0.73	+1.01	+0.103
0.083	uniform pooling, 130 tok.	206 MiB	−4.20	−1.95	−1.49	+22.87	−4.26	−2.95	+2.940
0.083	**AxisKV**, 4-bit recent	206 MiB	+0.10	−0.12	−0.06	−1.40	−1.13	+0.58	+0.213

Table 4: **Ablations at the operating point** (30 seconds, $n = 100$). *Keep vs. evict*: each distant slot kept as one token, minus the distant slots evicted with the recent window quantized to the same budget. *Correction removed*: the allocation without the analytic softmax correction of Tuncer et al. (2026), minus with it.

Comparison	Ratio	subject ↑	backgr. ↑	smooth. ↑	dynamic	aesth. ↑	imaging ↑	disc. ↓
keep vs. evict	8×	+0.85	+0.68	+0.47	+10.67	+0.93	+1.89	−0.713
	12×	+0.85	+0.62	+0.40	+12.93	+0.64	+1.44	−0.676
correction removed	8×	+0.00	−0.02	+0.01	−1.60	−0.08	−0.37	−0.046
	12×	+0.03	+0.04	+0.01	−0.07	+0.31	−0.04	−0.100

three consistency dimensions, uniform two-bit quantization by up to half a point and uniform pooling by as much as 3.35. The pairwise margins are $+0.60$ subject consistency and $+2.88$ imaging points over two-bit quantization ($t = 6.3$ and 6.5), and $+3.48$ and $+3.57$ over pooling ($t = 17.9$ and 8.1). At $\bar{M} = 0.083$, where uniform quantization has no operating point because it would need 1.33 bits, only pooling remains, and the margins grow to $+4.30$ and $+3.54$. And the allocation of **AxisKV** loses aesthetic quality relative to the uncompressed cache at every budget (-0.73 at $8\times$, -1.13 at $12\times$).

The two tables make the same point from opposite directions: against designs that keep more memory and against uniform schemes that keep the same, the allocation is the only one that leaves every consistency dimension at the level of the uncompressed cache. Appendix E shows frames from these rollouts, and Appendix G repeats the comparison at sixty seconds.

4.3 Ablation Study

Keep the distant slots. Keeping each distant slot as one token is better than evicting the distant slots and giving the recent window the same budget (Table 4). At $8\times$ it is worth $+0.85$ points of subject consistency, $+1.89$ of imaging quality and 0.71 of chunk discontinuity, and it raises dynamic degree by 10.67 points, the eviction having partially frozen the rollout exactly as it does under the probe (Section 3.2); at $12\times$ the margins are the same within noise. Eviction here also shortens the cache by six slots, which accounts for part of the loss; the attribution to the pinned slots themselves rests on Section 3.2, where removing three pinned slots costs quality and motion and removing three mid-lag slots, at the same cache size, does not.

The correction is needed only at the coarser budget. The analytic softmax correction that makes two-bit quantization usable (Appendix D.2) is not needed at $12\times$, where removing it moves no VBench dimension significantly and leaves chunk discontinuity slightly lower than with it. At $8\times$ it is worth 0.37 points of imaging quality, the only significant effect either way. A deployment at $12\times$

Table 5: **Ablations at the operating point** (30 seconds, $n = 100$). *Keep vs. evict*: each distant slot kept as one token, minus the distant slots evicted with the recent window quantized to the same budget. *Correction removed*: the allocation without the analytic softmax correction of Tuncer et al. (2026), minus with it.

Generator	Cache	subject ↑	backgr. ↑	smooth. ↑	dynamic	aesth. ↑	imaging ↑	disc. ↓
Self-Forcing	uncompressed	97.51	96.66	98.80	45.13	60.73	71.29	2.640
	uniform 2-bit (8×)	−0.47	−0.49	−0.14	−1.13	−1.60	−1.87	+0.226
	uniform pooling (8×)	−3.35	−1.67	−0.94	+22.27	−3.25	−2.56	+2.214
	AxisKV (8×)	+0.13	−0.06	−0.01	−1.20	−0.73	+1.01	+0.103
CausVid	uncompressed	97.35	96.65	98.13	49.00	57.05	66.75	2.940
	uniform 2-bit (8×)	−0.41	−0.20	−0.29	−3.87	−0.90	−1.47	+0.332
	uniform pooling (8×)	−2.50	−1.23	−0.41	+25.93	+0.55	−1.92	+1.490
	AxisKV (8×)	−0.23	−0.09	+0.30	−8.27	−5.38	−2.86	+0.139
Rolling Forcing	uncompressed	96.33	95.88	98.08	66.47	57.74	69.00	2.631
	uniform 2-bit (8×)	−0.11	−0.10	+0.06	−4.93	−0.24	−1.63	−0.026
	uniform pooling (8×)	−2.02	−0.95	−0.64	+12.47	+0.42	+0.18	+1.952
	AxisKV (8×)	+0.32	−0.02	+0.10	−4.60	−0.59	+0.96	−0.090
LongLive	uncompressed	98.14	97.07	98.98	35.40	61.98	70.81	2.403
	uniform 2-bit (8×)	−0.39	−0.38	−0.15	−0.67	−1.10	−0.99	+0.201
	uniform pooling (8×)	−1.68	−1.16	−0.69	+16.93	−2.68	−0.34	+1.543
	AxisKV (8×)	+0.01	−0.24	−0.06	−0.40	−0.57	+0.81	−0.042

can therefore drop the correction and use fused attention kernels; at $8\times$ it buys a third of an imaging point.

4.4 Transfer Across Generators

We apply the $8\times$ allocation unchanged to CausVid (Yin et al., 2025), Rolling Forcing (Liu et al., 2026) and LongLive (Yang et al., 2026a), three further generators derived from the same Wan2.1 prior. Table 5 reports the result. Read against each generator's own uncompressed rollout, the allocation transfers on three of the four. On Self-Forcing, Rolling Forcing and LongLive it leaves subject consistency within a quarter of a point of the uncompressed cache and is the only one of the three policies that *improves* imaging quality ($+1.01$, $+0.96$, $+0.81$), while uniform two-bit quantization costs imaging quality on every generator (-0.99 to -1.87) and uniform pooling costs one to three points of subject consistency everywhere. Pooling's dynamic degree rises by 12 to 26 points on all four generators while its consistency dimensions fall, the churn of Section 3.3 reappearing unchanged in every model. On CausVid the allocation is the most expensive of the three policies, losing 5.38 points of aesthetic quality and 2.86 of imaging quality against its own uncompressed rollout, where two-bit quantization loses 0.90 and 1.47. Appendix F explains that failure in detail.

5 Conclusion

We asked what each position in the KV cache of an autoregressive video generator contributes, by intervening on it persistently and at matched position rather than by reading attention weights. Positions differ in kind, not only in degree: distant slots must be present but little of their content is read, so each can be reduced to a single token, whereas the recent window must keep its spatial resolution yet tolerates two-bit precision. Compressing each position along the axis it tolerates leaves every VBench-Long dimension of Self-Forcing within 1.4 points of the uncompressed cache at $8\times$ and $12\times$, and beats uniform pooling at equal memory by three to four points of subject consistency. Its price is aesthetic quality, the one dimension on which it is consistently worse than the cache it compresses; once three-bit quantization fits the budget the uniform scheme is lossless everywhere and the allocation is unnecessary. The same probe says where to stop: it measures, on an unmodified generator, how much its pinned states carry, and on the one of four generators where that is largest our allocation is the most expensive policy rather than the cheapest. The contribution is thus a way to measure cache function causally, and the finding that a position's requirements are not predicted by its importance; all four generators descend from one video prior, and our memory figures are an accounting rather than a wall-clock measurement.

AI USE STATEMENT

We used generative AI tools in this work to discuss candidate hypotheses and help implement and debug the intervention, evaluation and analysis code, and to search for and summarize related work. On the writing side, they were used to improve the clarity and readability of the manuscript and to help format tables and figures. We did not use generative AI tools to generate experimental data, to compute or select the reported numbers, or to decide the claims of the paper. We have reviewed all AI-assisted work. AI-assisted code was read, modified where necessary, and verified by reproducing reported quantities from the raw per-video outputs. We take responsibility for the final content of this work, including all text, claims, figures, code and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work analyses and compresses the KV cache of existing, publicly released autoregressive video generators; it trains no new generative model and introduces no new generative capability. Its effect is to reduce the memory a long rollout requires, which makes long-form text-to-video generation cheaper to run. That cost reduction is dual-use: the same efficiency that makes long-horizon generation practical for research and creative work also lowers the barrier to producing misleading synthetic video at length. We encourage disclosure that video is synthetic wherever authenticity matters, and we note that our method neither adds nor removes any identity- or likeness-conditioning mechanism, so the safeguards appropriate to the underlying generators apply unchanged to their compressed versions.

The study involves no human subjects and no collection of personal or sensitive data. All prompts come from the public VBench prompt list (Huang et al., 2024), and all model weights are the publicly released checkpoints of Self-Forcing (Huang et al., 2026), CausVid (Yin et al., 2025), Rolling Forcing (Liu et al., 2026) and LongLive (Yang et al., 2026a), used for research in accordance with their respective licenses. Any bias or unsafe content present in those checkpoints or in the prompt list is inherited by the videos we generate; our interventions do not correct it, and we did not audit it. We release no new dataset.

REPRODUCIBILITY STATEMENT

Every experiment in this paper is an inference-time intervention on released checkpoints: nothing is trained, and no sampling hyper-parameter is changed, so a run is determined by the checkpoint, the prompt, the initial noise and the cache policy. Appendix A gives the generator configuration, the cache layout, the rollout lengths, the metric definitions and the exact specification of every compression arm. The 100 prompts are drawn once with a fixed seed from the public VBench list and then held fixed for every experiment, and each prompt's initial noise is drawn from a generator seeded by that prompt's index, so arms differ only in the cache policy and any two arms can be compared frame by frame. We will release the intervention and evaluation code together with the prompt list, the seeds and the configuration files for each arm, which is sufficient to regenerate the videos and reproduce every reported number.

REFERENCES

Hanmo Chen, Chenghao Xu, Xu Yang, Xuan Chen, and Cheng Deng. Past-and future-informed kv cache policy with salience estimation in autoregressive video diffusion. *arXiv preprint arXiv:2601.21896*, 2026a.

Jiayu Chen, Junbei Tang, Wenbiao Zhao, Maoliang Li, Jiayi Luo, Zihao Zheng, Jiawei Yang, Guojie Luo, and Xiang Chen. Pyramid forcing: Head-aware pyramid kv cache policy for high-quality long video generation. *arXiv preprint arXiv:2605.13111*, 2026b.

Shuo Chen, Cong Wei, Sun Sun, Ping Nie, Kai Zhou, Ge Zhang, Ming-Hsuan Yang, and Wenhu Chen. Context forcing: Consistent autoregressive video generation with long context. *arXiv preprint arXiv:2602.06028*, 2026c.

Yusuf Dalva and Pinar Yanardag. Adastate: Self-evolving anchors for streaming video generation. *arXiv preprint arXiv:2605.30349*, 2026.

Xiaomeng Fu, Jia Li, Yiming Hu, Yong Wang, Hayden Kwok-Hay So, Jiao Dai, Xiangxiang Chu, and Jizhong Han. Towards memory-efficient autoregressive video generation via instance-specific parametric absorption. In *European Conference on Computer Vision*, pp. 448–466. Springer, 2026.

Atticus Geiger, Zhengxuan Wu, Hanson Lu, Josh Rozner, Elisa Kreiss, Thomas Icard, Noah Goodman, and Christopher Potts. Inducing causal structure for interpretable neural networks. In *International Conference on Machine Learning*, pp. 7324–7338. PMLR, 2022.

Atticus Geiger, Duligur Ibeling, Amir Zur, Maheep Chaudhary, Sonakshi Chauhan, Jing Huang, Aryaman Arora, Zhengxuan Wu, Noah Goodman, Christopher Potts, et al. Causal abstraction: A theoretical foundation for mechanistic interpretability. *Journal of Machine Learning Research*, 26(83):1–64, 2025.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching. *arXiv preprint arXiv:2304.05969*, 2023.

Hang Guo, Zhaoyang Jia, Jiahao Li, Bin Li, Yuanhao Cai, Jiangshan Wang, Yawei Li, and Yan Lu. Efficient autoregressive video diffusion with dummy head. *arXiv preprint arXiv:2601.20499*, 2026.

Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*, 2024.

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

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, Yaohui Wang, Xinyuan Chen, Ying-Cong Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. doi: 10.1109/TPAMI.2025.3633890.

Yicheng Ji, Zhizhou Zhong, Jun Zhang, Qin Yang, XiTai Jin, Ying Qin, Wenhan Luo, Shuiyang Mao, Wei Liu, and Huan Li. Forcing-kv: Hybrid kv cache compression for efficient autoregressive video diffusion models. *arXiv preprint arXiv:2605.09681*, 2026.

Youngrae Kim, Qixin Hu, C-C Jay Kuo, and Peter Beerel. Memrope: Training-free infinite video generation via evolving memory tokens. In *European Conference on Computer Vision*, pp. 244–261. Springer, 2026.

Haodong Li, Shaoteng Liu, Zhe Lin, and Manmohan Chandraker. Rolling sink: Bridging limited-horizon training and open-ended testing in autoregressive video diffusion. *arXiv preprint arXiv:2602.07775*, 2026a.

Kunyang Li, Mubarak Shah, and Yuzhang Shang. Packcache: A training-free acceleration method for unified autoregressive video generation via compact kv-cache. *arXiv preprint arXiv:2601.04359*, 2026b.

Maximilian Li and Lucas Janson. Optimal ablation for interpretability. *Advances in Neural Information Processing Systems*, 37:109233–109282, 2024.

Wuyang Li, Wentao Pan, Po-Chien Luan, Yang Gao, and Alexandre Alahi. Stable video infinity: Infinite-length video generation with error recycling. In *International Conference on Learning Representations*, volume 2026, pp. 23406–23432, 2026c.

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

Yunhong Lu, Yanhong Zeng, Haobo Li, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jiapeng Zhu, Hengyuan Cao, Zhipeng Zhang, Xing Zhu, et al. Reward forcing: Efficient streaming video generation with rewarded distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34385–34397, 2026.

Jiayi Luo, Qiyan Liu, Tengyang Wang, JunHao Liu, Jiayu Chen, Cong Wang, Hanxin Zhu, Chen Gao, Xiaobin Hu, Qingyun Sun, et al. Future forcing: Future-aware training-free kv cache policy for autoregressive video generation. *arXiv preprint arXiv:2605.30083*, 2026.

Chengtao Lv, Yumeng Shi, Yushi Huang, Ruihao Gong, Shen Ren, and Wenya Wang. Light forcing: Accelerating autoregressive video diffusion via sparse attention. *arXiv preprint arXiv:2602.04789*, 2026.

Yuexiao Ma, Xuzhe Zheng, Jing Xu, Xiwei Xu, Feng Ling, Xiawu Zheng, Huafeng Kuang, Huixia Li, Xing Wang, Xuefeng Xiao, et al. Flow caching for autoregressive video generation. *arXiv preprint arXiv:2602.10825*, 2026.

Maisha Maliha and Dean F Hougen. Mechanistic interpretability of text-to-image diffusion models via cross-attention interventions. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25287–25299, 2026.

Xiaofeng Mao, Shaohao Rui, Kaining Ying, Bo Zheng, Chuanhao Li, Mingmin Chi, and Kaipeng Zhang. Packforcing: Short video training suffices for long video sampling and long context inference. In *European Conference on Computer Vision*, pp. 576–593. Springer, 2026.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Dvir Samuel, Issar Tzachor, Matan Levy, Michael Green, Gal Chechik, and Rami Ben-Ari. Fast-ar: Fast autoregressive video diffusion and world models with temporal cache compression and sparse attention. In *Forty-third International Conference on Machine Learning*.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, 2020.

Jiahao Tian, Yiwei Wang, Gang Yu, and Chi Zhang. Head forcing: Long autoregressive video generation via head heterogeneity. *arXiv preprint arXiv:2605.14487*, 2026.

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

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401, 2020.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng

Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Xiaofei Wu, Guozhen Zhang, Zhiyong Xu, Yuan Zhou, Qinglin Lu, and Xuming He. Pack and force your memory: Long-form and consistent video generation. *arXiv preprint arXiv:2510.01784*, 2025.

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

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations*, volume 2024, pp. 21875–21895, 2024.

Boxun Xu, Yuming Du, Zichang Liu, Siyu Yang, Ziyang Jiang, Siqi Yan, Rajasi Saha, Albert Pumarola, Wenchen Wang, and Peng Li. Sparse forcing: Native trainable sparse attention for real-time autoregressive diffusion video generation. *arXiv preprint arXiv:2604.21221*, 2026.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation. In *International Conference on Learning Representations*, volume 2026, pp. 90037–90059, 2026a.

Yang Yang, Tianyi Zhang, Wei Huang, Jinwei Chen, Boxi Wu, Xiaofei He, Deng Cai, Bo Li, and Peng-Tao Jiang. Anchor forcing: Anchor memory and tri-region rope for interactive streaming video diffusion. *arXiv preprint arXiv:2603.13405*, 2026b.

Hidir Yesiltepe, Jiazhen Hu, Tuna Han Salih Meral, Adil Kaan Akan, Kaan Oktay, Hoda Eldardiry, and Pinar Yanardag. Videomla: Low-rank latent kv cache for minute-scale autoregressive video diffusion. *arXiv preprint arXiv:2605.30351*, 2026a.

Hidir Yesiltepe, Tuna Meral, Adil Kaan Akan, Kaan Oktay, and Pinar Yanardag. Infinity-rope: Action-controllable infinite video generation emerges from autoregressive self-rollout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40256–40265, 2026b.

Jung Yi, Wooseok Jang, Paul Hyunbin Cho, Jisu Nam, Heeji Yoon, and Seungryong Kim. Deep forcing: Training-free long video generation with deep sink and participative compression. *arXiv preprint arXiv:2512.05081*, 2025.

Jung Yi, Minjae Kim, Paul Hyunbin Cho, Wooseok Jang, Sangdoo Yun, and Seungryong Kim. Worldkv: Efficient world memory with world retrieval and compression. *arXiv preprint arXiv:2605.22718*, 2026.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In *NeurIPS*, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024b.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22963–22974. IEEE, 2025.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, volume 2024, pp. 1651–1678, 2024.

Wenqu Zhao, Xuemin Chi, Xin Zhang, Guoqing Ma, Baorun Li, Jianjie Fang, Peizhi Tang, Chen Gao, and Wei Wu. Densitykv: Density-guided kv cache compression for long video generation. *arXiv preprint arXiv:2608.27922*, 2026a.

Zengqun Zhao, Yanzuo Lu, Ziquan Liu, Jifei Song, Jiankang Deng, and Ioannis Patras. Relax forcing: Relaxed kv-memory for consistent long video generation. *arXiv preprint arXiv:2603.21366*, 2026b.

Junhao Zhuang, Shiyi Zhang, Yuxuan Bian, Yaowei Li, Yawen Luo, Yijun Liu, Weiyang Jin, Songchun Zhang, Xianglong He, Xuying Zhang, et al. Self gradient forcing: Native long video extrapolation. *arXiv preprint arXiv:2607.20368*, 2026.