# EVICTION IS NOT FORGETTING: READ LONG, WRITE LOCAL IN STREAMING VIDEO DIFFUSION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Block-autoregressive video diffusion models generate long videos using bounded key/value (K/V) caches, yet visual consistency often deteriorates over time. We identify a source of this drift in the cache-writing process: attending to old K/V transfers historical information into new entries, allowing its influence to survive eviction. Under fixed video inputs and first-in-first-out retention, we establish a tight upper bound of $LW$ writes for the persistence of cache differences in an $L$-layer model with a $W$-block window, compared with $W$ writes when the writer excludes old K/V. Writing each block independently removes this propagation but omits temporal context across block boundaries. We introduce ARROW, a training-free method built on the principle of read long, write local. ARROW retains full access to cached history during denoising, while computing each new entry from the clean latents of the current and preceding blocks without old K/V. It stores only the current block's K/V, preserving boundary context without rewriting earlier entries. Across Self Forcing (Wan2.1), MAGI-1, and HunyuanVideo-1.5, ARROW improves VBench-Long quality by 3.6–6.2 points and LoCoT2V-Bench character consistency by 10.6–29.4 points, with 5–17% additional generation time. Controlled replay and ablations isolate the benefits of eliminating direct cache propagation and preserving boundary context, while confirming that long-range reading remains necessary.

## 1 INTRODUCTION

Block-autoregressive, or streaming, diffusion models such as CausVid (Yin et al., 2025), Self Forcing (Huang et al., 2025), MAGI-1 (Sand.ai et al., 2025) and LongLive (Yang et al., 2026) now support minute-long text-to-video generation. These models divide video latents into short blocks; Self Forcing (Huang et al., 2025) on Wan2.1 (Wan Team et al., 2025) uses three latent frames, or twelve output frames, per block. For every block the model performs three steps: it denoises the block while attending to a key/value (K/V) cache of earlier blocks, passes the finished clean latent through the network once more to compute the K/V stored for this block, and inserts these K/V into a fixed-size cache, evicting the oldest block when the cache is full. The cache is the only path by which the swimmer's appearance and action (Figure 1, top), the Shanghai skyline (Figure 1, middle), and the beach layout (Figure 1, bottom) reach the next block. A one-minute Wan video is written into the cache 81 times, and what each write stores decides whether these details persist at the end.

Research on long videos has improved two parts of this loop. Training methods change how the network responds to its own history, by training on self-generated rollouts (Huang et al., 2025; Po et al., 2026), extending rollouts to minutes (Cui et al., 2025; Liu et al., 2026; Yang et al., 2026) or optimizing rewards (Lu et al., 2026). Retention methods decide which entries stay in the cache, through attention sinks (Xiao et al., 2024), compression (Yi et al., 2026) and positional re-indexing (Yesiltepe et al., 2026). Most of them keep the second step, the writing pass, in the form of language-model decoding: Self Forcing and minWM compute a block's K/V while attending to the current cache (Huang et al., 2025; Zhao et al., 2026). With this design, a fixed window appears to limit memory: after enough further blocks an entry is evicted, and its content should influence later blocks only through what the video itself shows.

We find that eviction does not remove an old block's influence (Figure 2). During writing, one layer mixes old K/V into the new block, and the next stores K/V from that mixture. The new entry carries



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

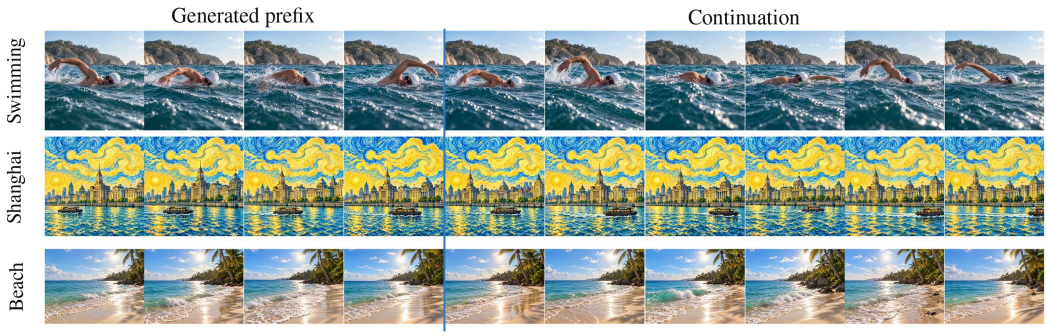



Figure 1: ARROW on Wan continuing three generated videos: four prefix frames left of the blue divider, six continuation frames right of it. Subjects and scene layout persist across the divider.

a copy of older entries after they are evicted, and the next write can copy it one layer higher. For identical video input, a cache difference can thus last for the number of layers times the window length, instead of just one window. For Wan the bound is 180 writes, more than twice a one-minute video. Suppressing copies for one window also improves subject and character consistency after all entries written during that period have left the cache.

A direct remedy is to write each block from its own clean latent alone, as MAGI-1 does (Sand.ai et al., 2025). This removes the copies, but the first frames of a block are then encoded without the motion or visibility change that leads into them: a hand that starts moving at the end of one block, or a car that enters the view, appears in the next block without its cause. On Wan, this current-only writer raises LoCoT2V character consistency from 36.4 to 52.5, and adding the previous block raises it to 65.8.

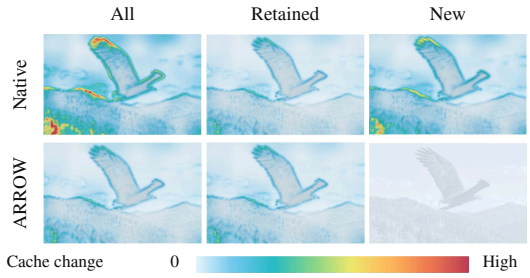



Figure 2: Eviction is not forgetting. Two runs write the same clean video into caches that differ at the start. Maps show the cache difference on one scale: over all entries (All), in the initial entries still in the window (Retained), and in entries written afterwards (New). The native writer copies the initial difference into new entries; ARROW's New map is near zero.

These two findings lead to one principle: *read long, write local*. Denoising should read all retained history to anchor appearance and layout. Each entry should be written from clean frames around its own block, so its dependence on the past comes through the video. ARROW (**A**utoregressive **R**ead with **R**aw-**O**verlap **W**rite) applies this principle to a frozen model. It writes from the clean latents of the preceding and current blocks under a causal mask without old K/V, stores only the current block's K/V, and keeps the preceding entry unchanged. Denoising, cache size, retention policy and positions stay native, so ARROW applies to block-autoregressive models with a separate writing pass. The persistent overhead is one latent block (below 1.5 MiB); the two-block writing pass adds 5–17% to generation time and 2.2–5.7 GiB of temporary peak memory.

On Self Forcing (Wan2.1-1.3B), MAGI-1-4.5B and HunyuanVideo-1.5-8B, ARROW improves VBench-Long quality by 3.6–6.2 points and LoCoT2V-Bench character consistency by 10.6–29.4 points. On Wan it also reaches the highest VBench-Long quality and LoCoT2V character consistency among seven long-video systems that retrain the same backbone or change its retention. The gain concentrates in the similarity between distant clips and is negligible for five-second videos, so it grows with the number of cache writes. Our contributions are:

- We show that the writing pass copies old K/V into new cache entries, so that eviction does not bound an old block's influence; we bound how long these copies persist and measure their harm with a replay intervention.
- We propose ARROW, a training-free writing rule that computes each entry from two adjacent clean blocks without old K/V and stores only the current block.
- On three models and two long-video benchmarks, we isolate each design choice: removing copied K/V, adding the previous block, keeping its stored entry and reading long history.

## 2 Related Work

**Autoregressive diffusion for long videos.** Video diffusion models denoise clips jointly with bidirectional attention (Ho et al., 2022; Polyak et al., 2024; Wan Team et al., 2025; Wu et al., 2025), typically over a fixed training horizon. Sliding-window schemes denoise frames at staggered noise levels (Kim et al., 2024; Ruhe et al., 2024). FIFO-Diffusion reuses pretrained models, whereas Rolling Diffusion requires training. Causal models generate frames or blocks using past context: Diffusion Forcing trains with independent per-frame noise levels using a recurrent architecture in its original video prediction experiments (Chen et al., 2024), CausVid distills a bidirectional teacher into a causal student (Yin et al., 2025), and MAGI-1 trains a chunk-wise model at scale (Sand.ai et al., 2025). Their central difficulty is error accumulation: a model trained on clean history conditions on its own imperfect outputs at test time (Bengio et al., 2015). Self Forcing therefore trains on self-generated rollouts (Huang et al., 2025), and later work extends training to minute-long rollouts, rolling denoising windows, rewards and aggregated corrective data (Cui et al., 2025; Liu et al., 2026; Yang et al., 2026; Lu et al., 2026; Po et al., 2026). These methods change how the network responds to its history and require training for each backbone. Most of them keep the decoding-style writing pass analyzed in Section 3. ARROW takes a checkpoint trained this way, such as Self Forcing, and changes what its cache stores.

**What the K/V cache stores.** Language-model caches grow with the sequence, so streaming methods keep a bounded subset: StreamingLLM retains initial sink tokens and a recent window (Xiao et al., 2024), and H2O retains tokens that receive high attention (Zhang et al., 2023). Video generators adapted these retention policies. LongLive and Reward Forcing keep sink frames (Yang et al., 2026; Lu et al., 2026), Deep Forcing adds deeper sinks and attention-based compression (Yi et al., 2026), Infinity-RoPE re-indexes temporal positions (Yesiltepe et al., 2026), FramePack compresses older frames into fewer tokens (Zhang et al., 2025), Ca2-VDM shares cached features across denoising steps (Gao et al., 2025), and SANA-Video keeps a constant-size linear-attention state (Chen et al., 2026b). These policies bound memory and choose which history stays readable. Fewer works change how an entry is computed. Block Diffusion computes a block's cache in a separate clean pass (Arriola et al., 2025). LongLive recomputes the cache from clean frames when the prompt changes, and BAgger refreshes K/V from a 12-latent-frame clean window (Yang et al., 2026; Po et al., 2026). Self Gradient Forcing trains the clean pass with future losses (Zhuang et al., 2026). KV Packet and DSCache build context-independent caches for documents and streamed video (Chen et al., 2026a; Pang et al., 2026). The remaining bottleneck lies in the writing pass itself: a retention policy can evict an entry, yet what the writer copied into later entries stays. ARROW fixes the input of every write and leaves retention to the host, so it composes with Deep Forcing and Infinity-RoPE (Section 5.6).

## 3 Why Eviction Is Not Forgetting

### 3.1 Block-Autoregressive Generation with a K/V Cache

Let $x_1, \dots, x_T$ be the clean latent blocks of a video, each with $N$ tokens, and let $\theta$ be a frozen network with $L$ attention layers. The cache entry of block $t$, $C_t = \{(K_t^\ell, V_t^\ell)\}_{\ell=1}^{L}$, holds its keys and values at every layer. We write $\mathcal{M}_t$ for the set of entries retained after block $t$ and $\mathcal{M}_t^\ell$ for its layer-$\ell$ part. Block $t$ is produced in three steps:

$$x_t = G_\theta\big(\epsilon_t,\, \mathcal{M}_{t-1}\big), \qquad C_t = U_\theta\big(x_t,\, \mathcal{M}_{t-1}\big), \qquad \mathcal{M}_t = \Pi\big(\mathcal{M}_{t-1}, C_t\big). \tag{1}$$

The denoising pass $G_\theta$ runs the native sampler from noise $\epsilon_t$ and reads the cache at every sampling step. The writing pass $U_\theta$ runs the network once on $x_t$ at the native clean noise level and returns the K/V of block $t$. The retention rule $\Pi$ inserts $C_t$. In its first-in-first-out form it evicts the oldest entry once $W$ blocks are stored, and hosts may add sink entries or compression. Both passes also receive the text embedding and the native temporal positions, which we omit from the notation. We call $U_\theta$ the writer and $G_\theta$ the reader. A current-only writer, such as that of MAGI-1, computes $U_\theta(x_t, \varnothing)$.

### 3.2 How Old K/V Enter a New Entry

Inside the writing pass, let $h_t^\ell$ be the hidden state of block $t$ at the input of layer $\ell$. The layer projects $h_t^\ell$ to the K/V that the cache stores, and then updates it by attention over the current block and the

cached history:

$$\big(K_t^\ell, V_t^\ell\big) = \mathcal{P}_\ell\big(h_t^\ell\big), \qquad h_t^{\ell+1} = B_\ell\big(h_t^\ell,\, \mathcal{M}_{t-1}^\ell\big). \tag{2}$$

Layer 1 thus stores K/V that depend on $x_t$ alone, while layer $\ell \geq 2$ stores K/V computed from a hidden state that has already attended to layers $1, \ldots, \ell - 1$ of the cache. The past therefore reaches a new entry by two routes. Differentiating the write in (1) with respect to the cache gives

$$\frac{\mathrm{d}C_t}{\mathrm{d}\mathcal{M}_{t-1}} = \underbrace{\frac{\partial U_\theta}{\partial x_t}\frac{\partial G_\theta}{\partial \mathcal{M}_{t-1}}}_{\text{through the generated block}} + \underbrace{\frac{\partial U_\theta}{\partial \mathcal{M}_{t-1}}}_{\text{copied from old K/V}}. \tag{3}$$

The first term records what block $t$ shows. The second records what the cache contained, independently of the video. The two routes treat history differently. The reader weighs an old entry against the current block while the entry is in the window, whereas a copy is stored under the entry and position of block $t$, is read by later blocks as part of block $t$, and stays for another $W$ writes after its source is evicted. Earlier errors thus return with a delay and outside the control of the retention rule.

**Theorem 1** (Lifetime of cache differences under fixed video)**.** *Run the writing and retention steps twice on the same clean blocks, starting from two different caches, with deterministic computation and first-in-first-out retention of $W$ blocks. If every entry is written without old K/V, the two caches coincide after at most $W$ writes. If an $L$-layer writer attends to old K/V as in (2), they coincide after at most $LW$ writes, and this bound is tight.*

The layer order in (2) explains the factor $L$. New layer-1 entries agree from the first write, so layer 1 of the two caches agrees after one window. New layer-2 entries then agree, and every further layer needs one more window (Appendix B.1). For Wan ($L = 30$, $W = 6$) the bound is 180 writes, and for Hunyuan ($L = 54$, $W = 5$) it is 270, whereas one-minute videos contain 81 and 61 blocks. Figure 2 shows the copies: after both runs write the same video, the native writer's new entries still differ, while ARROW's new entries coincide.

### 3.3 What the Current Block Alone Cannot Show

A current-only writer removes the copied term in (3) together with the frames just before the block. The first frames of a 0.75-second Wan block continue a movement, a camera pan or an occlusion that began in the previous block. A writer that sees only $x_t$ encodes them without the frames that explain them, although the next denoising pass reads $C_t$ to continue exactly this motion. These frames are available, since $x_{t-1}$ was completed one step earlier. A writing pass should therefore meet two requirements: clean video as its only input, so that the copied term vanishes, and a view of the transition into the block. Section 4 builds the writer with the smallest input that meets both.

## 4 ARROW: Read Long, Write Local

ARROW replaces the writing step in (1) and leaves denoising and retention native (Figure 3). The native, current-only and ARROW writers differ only in their input:

$$\begin{aligned} &\text{native:} && C_t = U_\theta\big(x_t,\, \mathcal{M}_{t-1}\big), \\ &\text{current-only:} && C_t = U_\theta\big(x_t,\, \varnothing\big), \\ &\text{ARROW:} && C_t = U_\theta\big([x_{t-1}, x_t],\, \varnothing\big)\big|_{x_t}, \end{aligned} \tag{4}$$

where $[x_{t-1}, x_t]$ concatenates two clean blocks along time and $\cdot|_{x_t}$ keeps the $N$ K/V rows of block $t$ at every layer. Each element of the ARROW rule addresses one requirement from Section 3.

**Write from clean latents, without old K/V.** The empty cache $\varnothing$ removes the copied term in (3). The previous block enters as its clean latent $x_{t-1}$, kept in a one-block buffer, and its hidden states are recomputed inside the pass. Attending to the stored entry $C_{t-1}$ instead would make each entry depend on the one before it, and the copies would again chain back through the whole video.

**Add exactly one previous block.** The pair $[x_{t-1}, x_t]$ is the smallest input that contains the transition into block $t$. Under the block-causal mask $\mathbf{A}_2 = \left(\begin{smallmatrix} 0 & -\infty \\ 0 & 0 \end{smallmatrix}\right)$, block $t-1$ attends to itself and block $t$ attends to both, so the K/V of block $t$ are computed with the transition in view. Older frames remain

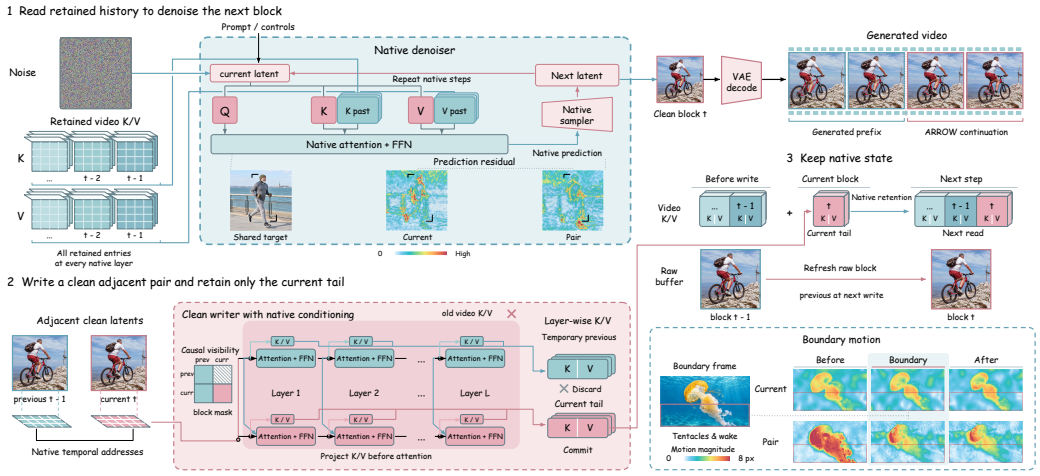



Figure 3: ARROW. (1) Denoising reads all retained K/V natively. (2) The writer processes the clean latents of blocks $t-1$ and $t$ under a block-causal mask without old video K/V and commits only the K/V of block $t$ (the current tail). (3) Cache size, positions and retention stay native. Insets: prediction residual and boundary motion with current-only and pair entries.

available to denoising through the cache. In our experiments a third block changes quality by at most 0.04 points and adds 7–14% generation time (Section 5.4). Unlike refreshing K/V from a long clean window, as in BAgger (Po et al., 2026), its cost does not grow with the cache size. The first block is written from $x_1$ alone.

**Store only the current block.** The pair pass also yields K/V for block $t-1$. Inside the pass this block sees only itself, so these K/V are a current-only encoding. The entry already stored for block $t-1$ was written one step earlier from $[x_{t-2}, x_{t-1}]$ and contains the transition into block $t-1$. The two entries encode the same clean latent, yet they differ over the whole frame, including static sand (Figure 4): the context of a write changes the entry as a whole. ARROW therefore discards the recomputed K/V and keeps the stored entry. Each entry is thus computed once, from its own block and the preceding one, and never modified. With first-in-first-out retention, $\mathcal{M}_t$ becomes a function of the clean blocks $x_{t-W}, \ldots, x_t$, and an evicted block influences later entries only through the video.

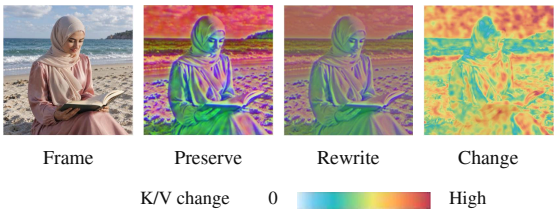



Figure 4: Two entries of the same block: stored (Preserve) and recomputed in the next pair pass (Rewrite), with their K/V difference (Change).

**Keep reading long.** The reader stays native: it attends to all retained entries, which keep the native layout, noise level and positions, and thus still sees earlier views of the subject and scene that the pair lacks (Section 5.6).

**Cost.** ARROW keeps the native cache size and adds one clean latent block (0.4–1.5 MiB). The pair doubles the video tokens of the writing pass, and per-block work stays constant over the video (Algorithm 1, Appendix B.3).

## 5 Experiments

### 5.1 Setup

**Models.** We apply ARROW to three frozen block-autoregressive models: Self Forcing on Wan2.1-T2V-1.3B (Huang et al., 2025; Wan Team et al., 2025), distilled MAGI-1-4.5B (Sand.ai et al., 2025), and the HunyuanVideo-1.5-8B generator of minWM (Wu et al., 2025; Zhao et al., 2026), abbreviated Wan, MAGI and Hunyuan (Self Forcing, MAGI-1 and minWM in figures. Vanilla denotes native).

Table 1: **Long-video quality and system cost.** Native–ARROW pairs are matched within each architecture. The seven external Wan methods retain their own system settings. VBench Long uses a 60-second request. LoCo uses its reference durations. Quality metrics are on 100-point scales. Time and allocated peak are per-video means on VBench Long. Sub./Bg.: subject/background consistency. DD/MS: Dynamic Degree/Motion Smoothness. PQ/OA/FGA: perceptual quality/overall alignment/fine-grained alignment. CC/BC: character/background consistency. HERD: Human Expectation Realization Degree. Bold marks the best displayed value within each architecture. Blue identifies ARROW in system comparisons.

| Method | VBench Long ↑ | | | | | | LoCoT2V-Bench ↑ | | | | | | Resources ↓ | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Qual. | Sem. | Sub. | Bg. | DD | MS | PQ | OA | FGA | CC | BC | HERD | Time (s) | GiB |
| *Wan2.1-1.3B / Self Forcing (Huang et al., 2025; Wan Team et al., 2025)* | | | | | | | | | | | | | | |
| Native | 78.98 | 73.94 | 88.54 | 91.92 | 44.0 | 98.20 | 73.7 | 44.2 | 27.1 | 36.4 | 98.1 | 70.9 | 270.8 | 15.0 |
| Deep Forcing (Yi et al., 2026) | 82.65 | 77.79 | 93.04 | 94.23 | 60.9 | 98.31 | 77.0 | 53.8 | 36.0 | 54.1 | 98.8 | 78.0 | 292.3 | 15.0 |
| Infinity-RoPE (Yesiltepe et al., 2026) | 83.33 | 78.58 | 94.31 | 94.92 | 56.1 | **98.84** | 78.0 | 55.7 | 38.5 | 56.5 | 99.0 | 80.0 | **215.3** | **11.0** |
| Rolling Forcing (Liu et al., 2026) | 82.00 | 73.06 | 93.61 | 94.45 | 39.1 | 98.64 | 78.0 | 51.6 | 32.6 | 57.5 | 98.9 | 79.4 | 506.9 | 13.7 |
| LongLive (Yang et al., 2026) | 83.22 | 79.41 | 94.72 | 95.61 | 47.3 | 98.72 | **80.7** | 55.5 | 36.0 | 55.4 | **99.2** | 81.3 | 225.4 | 12.6 |
| Self-Forcing++ (Cui et al., 2025) | 83.76 | 79.53 | 94.29 | 95.10 | 60.0 | 98.43 | 78.9 | 59.0 | 41.3 | 60.4 | 99.0 | 82.6 | 271.0 | 15.0 |
| Reward (Lu et al., 2026) | 84.30 | 79.65 | 93.90 | 94.65 | 71.1 | 98.70 | 78.3 | 58.7 | 40.2 | 57.5 | 98.8 | 80.8 | 234.5 | 12.6 |
| BAgger (Po et al., 2026) | 81.06 | 73.12 | 88.78 | 91.55 | **75.7** | 98.55 | 73.1 | 49.4 | 32.1 | 42.8 | 98.4 | 74.4 | 2899.3 | 23.5 |
| **ARROW** | **85.15** | **81.12** | **95.18** | **96.04** | 66.8 | 98.60 | 79.3 | **63.1** | **44.5** | **65.8** | 99.1 | **84.3** | 301.9 | 17.2 |
| *MAGI-1-4.5B, distilled (Sand.ai et al., 2025)* | | | | | | | | | | | | | | |
| Native | 75.43 | 64.89 | 83.01 | 87.70 | 43.3 | **99.12** | 72.0 | 42.6 | 29.7 | 38.6 | 98.4 | 73.8 | **6835.3** | **36.9** |
| **ARROW** | **80.04** | **70.52** | **89.98** | **93.36** | **52.5** | 99.00 | **74.9** | **52.4** | **37.7** | **49.2** | **98.9** | **79.3** | 7195.9 | 40.6 |
| *HunyuanVideo1.5-8B / minWM (Wu et al., 2025; Zhao et al., 2026)* | | | | | | | | | | | | | | |
| Native | 82.01 | 77.27 | 90.66 | 94.14 | 51.6 | 98.68 | 80.2 | 58.5 | 39.4 | 50.2 | 98.6 | 82.2 | **922.5** | **44.4** |
| **ARROW** | **85.57** | **81.96** | **95.41** | **96.60** | **62.1** | **98.83** | **82.1** | **65.3** | **46.0** | **62.0** | **99.2** | **85.7** | 1075.4 | 50.1 |

Wan and Hunyuan natively write with old K/V. MAGI natively writes each block alone, so its comparison isolates the value of the previous block. Within each Native–ARROW pair, weights, prompts, noise, sampler, cache size and positions are identical, and only the writing pass differs. On Wan we also compare seven long-video systems built on the same backbone, each with its own training and inference settings (Appendix C.2).

**Evaluation.** VBench-Long uses the 946 official records with 60-second requests (Huang et al., 2024; 2026), and LoCoT2V-Bench uses all 234 prompts at their reference durations of 30–64 seconds (Zheng et al., 2026). Confidence intervals come from 10,000 paired bootstrap resamples over prompts (Appendices C.3–C.4). The experiments answer five questions: whether ARROW improves long videos and where (Section 5.2), whether copied K/V persist after eviction and harm generation (Section 5.3), why the writer uses a pair and stores only the current block (Section 5.4), whether the frozen reader uses the added frames (Section 5.5), and whether reading should stay long (Section 5.6).

### 5.2 ARROW Improves Continuity across Blocks

**Matched comparisons.** ARROW improves every VBench-Long aggregate and LoCoT2V character consistency in all three models (Table 1). Wan gains 6.17 quality points (95% CI 6.09–6.25) and 29.43 character-consistency points (28.95–29.92). MAGI gains 4.61 and 10.6, and Hunyuan 3.56 and 11.8. Wan and Hunyuan benefit from removing the copies and from the boundary frames, whereas MAGI, whose native writer already excludes old K/V, benefits from the boundary frames alone. On Wan, subject and background consistency rise by 6.6 and 4.1 points. With character consistency, they measure whether the same person and scene persist. Dynamic Degree rises as well, from 44.0 to 66.8 on Wan: the pair entry carries the ongoing motion into the next block, so ARROW videos stay consistent while moving more. Figure 5 shows both effects at 0, 20, 40 and 60 seconds.

**External systems.** Without retraining, ARROW on Wan obtains the best VBench-Long quality, semantic, subject and background scores and the best LoCoT2V alignment, character consistency and HERD among the nine Wan systems. Systems that retrain the model or change its retention lead on criteria close to their design: LongLive on LoCoT2V perceptual quality and background consistency, Infinity-RoPE on motion smoothness, and BAgger and the motion-rewarded Reward Forcing on Dynamic Degree. ARROW adds 11.5%, 5.3% and 16.6% generation time on Wan, MAGI and Hunyuan, and 2.2–5.7 GiB of temporary peak memory.

**Where the gain appears.** VBench-Long scores similarity within and between the clips of each video (Figure 6). Within-clip similarity is high for both writers. Between-clip similarity compares




Figure 5: One-minute generations on Wan at 0, 20, 40 and 60 s. Yellow boxes mark the rider's helmet and the dog's face. Compare the vehicle, the drum kit and the background along each row.

content separated by many cache writes. It is higher with ARROW for nearly all prompts and more than ten times higher where the native model drifts most. Five-second requests involve only five to seven writes, and their paired quality differences are 0.01–0.03 points with intervals containing zero (Appendix D.4). ARROW's effect thus grows with the number of writes.

### 5.3 Copied K/V Persist after Eviction and Harm Generation

Theorem 1 bounds how long copies can last. This experiment tests whether copies that outlive their source change the generated video. We start from a full cache taken from a reference video. Two branches write the next $W$ reference blocks from the current block alone. One branch attends to old K/V during these writes, as the native writer does, and the other does not (prefix-present and prefix-absent in Table 2). Both branches then write $W$ further reference blocks with the native writer, which evicts every entry from the intervention period, and finally generate freely from common noise with native writing and reading. The two branches see identical video until release, so any difference in the continuation comes from K/V copied into entries written after the intervention.

Table 2: **Fixed-input intervention.** Mean effect of prefix-absent minus prefix-present writing (points), after intervention-entry eviction and common free generation. Sub./Bg.: subject/background. TQ/CC: temporal quality/character consistency. Neither branch uses raw overlap. Bold marks the larger source-conditioned mean in each architecture.

| Reference | VBench Long | | LoCo | |
|---|---|---|---|---|
| | Sub. | Bg. | TQ | CC |
| *Wan2.1-1.3B (Huang et al., 2025; Wan Team et al., 2025)* | | | | |
| Native | **2.24** | **1.31** | **1.47** | **3.46** |
| ARROW | 1.62 | 0.92 | 1.32 | 3.11 |
| *HunyuanVideo1.5-8B (Wu et al., 2025; Zhao et al., 2026)* | | | | |
| Native | **1.54** | **0.92** | **1.30** | **3.19** |
| ARROW | 1.24 | 0.79 | 1.02 | 2.55 |

Removing old K/V during the intervention improves every metric in Table 2, in both models and from both reference sources. On Wan with native reference videos, character consistency rises by 3.46 points (95% CI 2.97–3.96). The effect of $W$ writes thus outlasts the entries written in them, as the $LW$ bound of Theorem 1 allows. The effect is larger on native than on ARROW reference videos in all eight cells. Native videos drift more (Table 1), so their caches hold more content that disagrees with the current frames. The larger effect is consistent with copies carrying this content forward. Likewise, the effects on Hunyuan, whose native videos drift less (character consistency 50.2 versus 36.4), are smaller than on Wan although its bound $LW$ is longer. Per prompt, subject-consistency gains are positive for nearly all prompts, while identity gains vary more (Figure 10, Appendix C.5). The intervention removes copies during only $W$ writes. A native writer copies at every one of the 61–81 writes of a video, and Section 5.4 measures the full difference between native and copy-free writing.

### 5.4 Why a Pair, and Why Store Only the Current Block

Table 3 fixes denoising and varies only the writer, so each element of (4) appears as a separate step. Removing old K/V (native to current-only) raises character consistency from 36.4 (Table 1) to 52.5 on Wan and from 50.2 to 55.0 on Hunyuan. MAGI's native writer is already current-only, so this step changes nothing there.

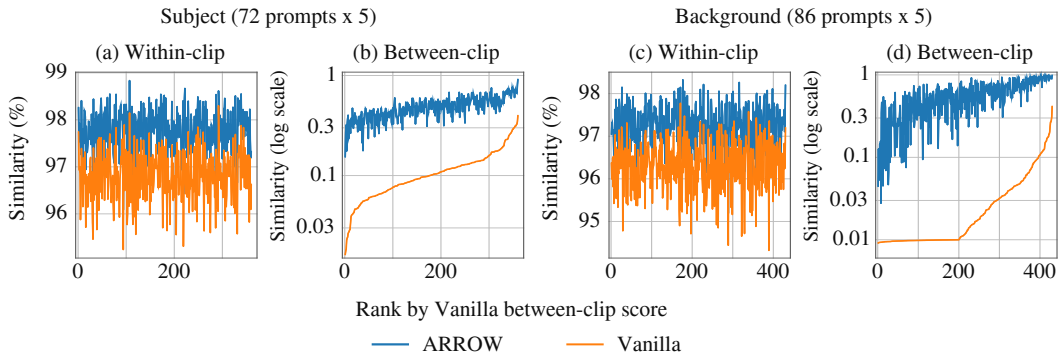



Figure 6: The gain lies between clips. VBench subject and background consistency on Wan per generation, sorted by the native (Vanilla) between-clip score. Between-clip similarity is on a log scale.

Table 3: **Writer ablations.** Native historical reading is fixed. Pair is ARROW. 3 blocks adds one predecessor. Rewrite also overwrites the resident predecessor. Q, DD and CC follow Table 1. Time (s) and allocated peak (GiB) are per-video means. Bold marks within-architecture optima. Ties are retained.

| Writer | VBench ↑ | | LoCo ↑ | Cost ↓ | |
|---|---|---|---|---|---|
| | Quality | DD | CC | Time | GiB |
| *Wan2.1-1.3B (Huang et al., 2025; Wan Team et al., 2025)* | | | | | |
| Current only | 82.25 | 59.5 | 52.5 | **270.4** | **15.8** |
| Pair | **85.15** | **66.8** | 65.8 | 301.9 | 17.2 |
| 3 blocks | 85.13 | **66.8** | **66.0** | 343.1 | 19.0 |
| Rewrite prev. | 83.49 | 63.7 | 59.5 | 303.1 | 17.2 |
| *MAGI-1-4.5B (Sand.ai et al., 2025)* | | | | | |
| Current only | 75.43 | 43.3 | 38.6 | **6835.3** | **36.9** |
| Pair | 80.04 | 52.5 | **49.2** | 7195.9 | 40.6 |
| 3 blocks | **80.08** | **52.7** | **49.2** | 7676.4 | 43.2 |
| Rewrite prev. | 78.49 | 51.2 | 45.8 | 7197.7 | 40.6 |
| *HunyuanVideo1.5-8B (Wu et al., 2025; Zhao et al., 2026)* | | | | | |
| Current only | 83.48 | 56.5 | 55.0 | **912.4** | **47.0** |
| Pair | **85.57** | **62.1** | **62.0** | 1075.4 | 50.1 |
| 3 blocks | 85.55 | 61.9 | 61.9 | 1223.4 | 53.7 |
| Rewrite prev. | 84.28 | 59.6 | 57.7 | 1078.6 | 50.1 |

Adding the previous block (current-only to pair) raises it further to 65.8, 49.2 and 62.0 on Wan, MAGI and Hunyuan. On MAGI this step is the entire gain. On Wan the two steps contribute similar amounts, 16.1 and 13.3 points. A third block changes every metric by at most 0.2 points and adds 7–14% time, so the benefit comes from the transition into the block, and older frames add only cost. Overwriting the previous block's stored entry with its recomputed K/V lowers character consistency by 6.3, 3.4 and 4.3 points at equal cost. The stored entry therefore holds information that the recomputed one lacks, namely the transition into block $t-1$. Figure 4 shows the two entries of one block (see also Appendix D.1).

### 5.5 The Frozen Reader Uses the Boundary Frames

Table 4: **Shared-reader prediction.** Mean MSE ($10^{-2}$, lower is better), with shared target and reader inputs at $t = 2, \ldots, T-1$. Native/ARROW identify reference trajectories. Bold marks within-host optima. Appendix C.5 specifies native targets.

| Writer | VBench Long ↓ | | LoCo ↓ | |
|---|---|---|---|---|
| | Native | ARROW | Native | ARROW |
| *Wan2.1-1.3B (Huang et al., 2025; Wan Team et al., 2025)* | | | | |
| 1 block | 19.01 | 16.60 | 23.64 | 20.71 |
| 2 blocks | **16.71** | **15.10** | **20.25** | **18.31** |
| *MAGI-1-4.5B (Sand.ai et al., 2025)* | | | | |
| 1 block | 28.29 | 26.69 | 32.21 | 29.36 |
| 2 blocks | **25.09** | **24.50** | **28.29** | **26.47** |
| *HunyuanVideo1.5-8B (Wu et al., 2025; Zhao et al., 2026)* | | | | |
| 1 block | 11.70 | 10.69 | 14.33 | 12.80 |
| 2 blocks | **10.29** | **9.69** | **12.33** | **11.20** |

Long-video metrics include every downstream change of the trajectory. To test the entry itself, we fix a reference video, the rest of the cache and the noisy next block, supply either the current-only or the pair entry for block $t$, and measure the error of the frozen network's prediction for block $t+1$. The pair entry lowers the error in all twelve conditions of Table 4, by 8–14%. On Wan with native VBench videos it falls from 19.01 to 16.71. Since all other inputs are shared, this difference measures the predictive information added by the previous block together with the change in how well the frozen network reads the entry (Proposition 1, Appendix B.2). The cache already holds an entry for block $t-1$, so the reduction shows that the reader predicts better when the transition is encoded inside the entry of block $t$. Reductions are larger on LoCoT2V, whose prompts describe more complex events, and on native reference videos. Figure 7 shows where the reduction occurs. With the current-only entry, the error concentrates on the jogger's outline and on the car's body and wheels, the parts that move across the block boundary. The pair entry lowers

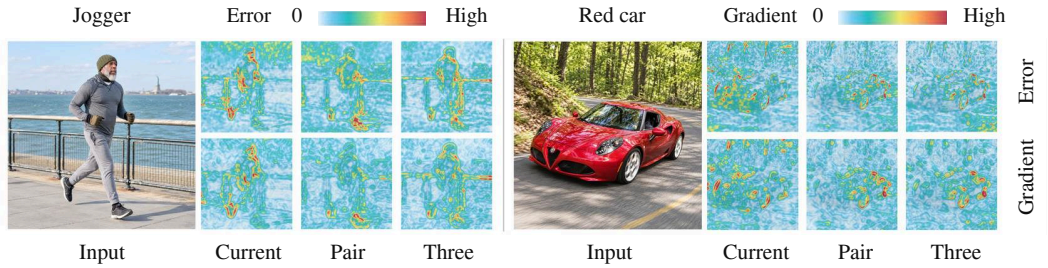



Figure 7: Prediction error with one (Current), two (Pair) and three (Three) clean blocks in the written entry. Targets and other reader inputs are shared. Error: residual. Gradient: its spatial variation.

the error in these regions. A third block lowers the error by at most a further 0.30 in the same units (Table 6), matching its small effect on video quality.

### 5.6 Reading Long History Remains Necessary

Since local writing helps, we test whether reading should also be local: if the pair entry carried all useful history, restricting the reader to the previous block would cost nothing. We cross the native and ARROW writers with two readers in Deep Forcing and Infinity-RoPE (Yi et al., 2026; Yesiltepe et al., 2026). The full reader uses all retained entries. Once the cache is full, the adjacent reader masks every retained entry older than the previous block, sink entries included. Both hosts otherwise keep their own retention policies.




Figure 8: With the ARROW writer fixed, the full reader keeps the station's roof and windows seen earlier. The adjacent reader alters them.

Table 5: **Writer–reader comparison.** All/Adjacent use all retained history or only the predecessor after native retention. Column pairs compare Native and ARROW writers in the same host. Q/CC follow Table 1. Bold selects reader access within each writer. No cross-host ranking is implied.

| Reader | VBench Quality ↑ | | LoCo CC ↑ | |
|---|---|---|---|---|
| | Native | ARROW | Native | ARROW |
| *Deep Forcing (Yi et al., 2026) / Wan2.1-1.3B* | | | | |
| All | **82.65** | **85.57** | **54.1** | **66.9** |
| Adjacent | 81.44 | 84.28 | 48.2 | 62.2 |
| *Infinity-RoPE (Yesiltepe et al., 2026) / Wan2.1-1.3B* | | | | |
| All | **83.33** | **85.16** | **56.5** | **66.1** |
| Adjacent | 82.33 | 84.14 | 52.4 | 62.4 |

Both factors matter, and their effects add up (Table 5). ARROW improves quality by 1.8–2.9 points under either reader, and the full reader improves quality by 1.0–1.3 points under either writer. With ARROW, full reading adds 4.65 character-consistency points in Deep Forcing (95% CI 4.19–5.11) and 3.67 in Infinity-RoPE (3.18–4.16). The reader effect on quality is nearly equal for both writers (1.21 vs. 1.29 points in Deep Forcing, 1.00 vs. 1.02 in Infinity-RoPE), so the two choices supply different information: the pair gives the transition into each block, and older entries give earlier views of the subject and scene. Figure 8 shows such a case. ARROW also composes with both retention policies: Deep Forcing with ARROW reaches 85.57 quality and 66.9 character consistency, above every Wan system in Table 1.

## 6 Conclusion

In block-autoregressive video diffusion, when the writing pass attends to old K/V, every new entry carries a copy of older entries, and a fixed window no longer bounds how long an old block influences generation. ARROW writes each entry from the clean latents of its own block and the block before it, stores only the current block, and keeps long-range reading for denoising. This single change to the writing input improves long-video consistency in three frozen models from 1.3B to 8B parameters at a small, constant cost, and it composes with retention policies that keep sinks or compress history. The result separates two choices that streaming models usually make together: how much history the reader attends to, which can grow with the memory budget, and what the writer sees, which should stay local.

## REFERENCES


Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block Diffusion: Interpolating Between Autoregressive and Diffusion Language Models. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2503.09573.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Hritik Bansal, Clark Peng, Yonatan Bitton, Roman Goldenberg, Aditya Grover, and Kai-Wei Chang. VideoPhy-2: A Challenging Action-Centric Physical Commonsense Evaluation in Video Generation. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2503.06800.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. In *NeurIPS*, volume 28, 2015. URL https://arxiv.org/abs/1506.03099.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment Anything with Concepts. *arXiv preprint arXiv:2511.16719*, 2025. URL https://arxiv.org/abs/2511.16719.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token Prediction Meets Full-Sequence Diffusion. In *NeurIPS*, volume 37, pp. 24081–24125, 2024. URL https://arxiv.org/abs/2407.01392.

Chuangtao Chen, Grace Li Zhang, Xunzhao Yin, Cheng Zhuo, Bing Li, and Ulf Schlichtmann. KV Packet: Recomputation-Free Context-Independent KV Caching for LLMs. *arXiv preprint arXiv:2604.13226*, 2026a. URL https://arxiv.org/abs/2604.13226.

Junsong Chen, Yuyang Zhao, Jincheng Yu, Ruihang Chu, Junyu Chen, Shuai Yang, Xianbang Wang, Yicheng Pan, Daquan Zhou, Huan Ling, Haozhe Liu, Hongwei Yi, Hao Zhang, Muyang Li, Yukang Chen, Han Cai, Sanja Fidler, Ping Luo, Song Han, and Enze Xie. SANA-Video: Efficient Video Generation with Block Linear Diffusion Transformer. In *International Conference on Learning Representations*, 2026b. URL https://arxiv.org/abs/2509.24695.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-Forcing++: Towards Minute-Scale High-Quality Video Generation. *arXiv preprint arXiv:2510.02283*, 2025. URL https://arxiv.org/abs/2510.02283.

Kaifeng Gao, Jiaxin Shi, Hanwang Zhang, Chunping Wang, Jun Xiao, and Long Chen. Ca2-VDM: Efficient Autoregressive Video Diffusion Model with Causal Generation and Cache Sharing. In *ICML*, volume 267 of *PMLR*, pp. 18550–18565, 2025. URL https://arxiv.org/abs/2411.16375.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video Diffusion Models. In *NeurIPS*, volume 35, 2022. URL https://arxiv.org/abs/2204.03458.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the Train-Test Gap in Autoregressive Video Diffusion. In *NeurIPS*, volume 38, pp. 167283–167308, 2025. URL https://arxiv.org/abs/2506.08009. Main Conference.

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 *CVPR*, pp. 21807–21818, 2024. URL https://arxiv.org/abs/2311.17982.

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. *TPAMI*, 48(3):3268–3285, 2026. doi: 10.1109/TPAMI.2025.3633890. URL https://arxiv.org/abs/2411.13503.

Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. FIFO-Diffusion: Generating Infinite Videos from Text without Training. In *NeurIPS*, volume 37, 2024. URL https://arxiv.org/abs/2405.11473.

Xinran Ling, Chen Zhu, Meiqi Wu, Hangyu Li, Xiaokun Feng, Cundian Yang, Aiming Hao, Jiashu Zhu, Jiahong Wu, and Xiangxiang Chu. VMBench: A Benchmark for Perception-Aligned Video Motion Generation. In *ICCV*, pp. 13087–13098, 2025. URL https://arxiv.org/abs/2503.10076.

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*, 2026. URL https://arxiv.org/abs/2509.25161.

Yunhong Lu, Yanhong Zeng, Haobo Li, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jiapeng Zhu, Hengyuan Cao, Zhipeng Zhang, Xing Zhu, Yujun Shen, and Min Zhang. Reward Forcing: Efficient Streaming Video Generation with Rewarded Distribution Matching Distillation. In *CVPR*, pp. 34385–34397, 2026. URL https://arxiv.org/abs/2512.04678.

Zhanzhong Pang, Dibyadip Chatterjee, Fadime Sener, and Angela Yao. Decouple and Cache: KV Cache Construction for Streaming Video Understanding. In *ICML*, 2026. URL https://arxiv.org/abs/2605.01858.

Ryan Po, Eric Ryan Chan, Changan Chen, and Gordon Wetzstein. BAgger: Backwards Aggregation for Mitigating Drift in Autoregressive Video Diffusion Models. In *CVPR*, pp. 43727–43739, 2026. URL https://arxiv.org/abs/2512.12080.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Kiran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali Thabet, Artsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie Gen: A Cast of Media Foundation Models. *arXiv preprint arXiv:2410.13720*, 2024. URL https://arxiv.org/abs/2410.13720.

David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling Diffusion Models. In *ICML*, volume 235 of *PMLR*, 2024. URL https://arxiv.org/abs/2402.09470.

Sand.ai, Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, W. Q. Zhang, Weifeng Luo, Xiaoyang Kang, Yuchen Sun, Yue Cao, Yunpeng Huang, Yutong Lin, Yuxin Fang, Zewei Tao, Zheng Zhang, Zhongshu Wang, Zixun Liu, Dai Shi, Guoli Su, Hanwen Sun, Hong Pan, Jie Wang, Jiexin Sheng, Min Cui, Min Hu, Ming Yan, Shucheng Yin, Siran Zhang, Tingting Liu, Xianping Yin, Xiaoyu Yang, Xin Song, Xuan Hu, Yankai Zhang, and Yuqiao Li. MAGI-1: Autoregressive Video Generation at Scale. *arXiv preprint arXiv:2505.13211*, 2025. URL https://arxiv.org/abs/2505.13211.

Kaiyue Sun, Kaiyi Huang, Xian Liu, Yue Wu, Zihan Xu, Zhenguo Li, and Xihui Liu. T2V-CompBench: A Comprehensive Benchmark for Compositional Text-to-video Generation. In *CVPR*, pp. 8406–8416, 2025. URL https://arxiv.org/abs/2407.14505.

Wan Team, 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. URL https://arxiv.org/abs/2503.20314.

Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, Linus, Patrol, Peizhen Zhang, Peng Chen, Penghao Zhao, Qi Tian, Songtao Liu, Weijie Kong, Weiyan Wang, Xiao He, Xin Li, Xinchi Deng, Xuefei Zhe, Yang Li, Yanxin Long, Yuanbo Peng, Yue Wu, Yuhong Liu, Zhenyu Wang, Zuozhuo Dai, Bo Peng, Coopers Li, Gu Gong, Guojian Xiao, Jiahe Tian, Jiaxin Lin, Jie Liu, Jihong Zhang, Jiesong Lian, Kaihang Pan, Lei Wang, Lin Niu, Mingtao Chen, Mingyang Chen, Mingzhe Zheng, Miles Yang, Qiangqiang Hu, Qi Yang, Qiuyong Xiao, Runzhou Wu, Ryan Xu, Rui Yuan, Shanshan Sang, Shisheng Huang, Siruis Gong, Shuo Huang, Weiting Guo, Xiang Yuan, Xiaojia Chen, Xiawei Hu, Wenzhi Sun, Xiele Wu, Xianshun Ren, Xiaoyan Yuan, Xiaoyue Mi, Yepeng Zhang, Yifu Sun, Yiting Lu, Yitong Li, You Huang, Yu Tang, Yixuan Li, Yuhang Deng, Yuan Zhou, Zhichao Hu, Zhiguang Liu, Zhihe Yang, Zilin Yang, Zhenzhi Lu, Zixiang Zhou, and Zhao Zhong. HunyuanVideo 1.5 Technical Report. *arXiv preprint arXiv:2511.18870*, 2025. URL https://arxiv.org/abs/2511.18870.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient Streaming Language Models with Attention Sinks. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.17453.

Chunyu Xie, Bin Wang, Fanjing Kong, Jincheng Li, Dawei Liang, Ji Ao, Dawei Leng, and Yuhui Yin. FG-CLIP 2: A Bilingual Fine-grained Vision-Language Alignment Model. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2510.10921.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, Song Han, and Yukang Chen. LongLive: Real-time Interactive Long Video Generation. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2509.22622.

Hidir Yesiltepe, Tuna Han Salih Meral, Adil Kaan Akan, Kaan Oktay, and Pinar Yanardag. Infinity-RoPE: Action-Controllable Infinite Video Generation Emerges From Autoregressive Self-Rollout. In *CVPR*, pp. 40256–40265, 2026. URL https://arxiv.org/abs/2511.20649.

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. In *ICML*, 2026. URL https://arxiv.org/abs/2512.05081.

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 *CVPR*, pp. 22963–22974, 2025. URL https://arxiv.org/abs/2412.07772.

Zhiyuan You, Xin Cai, Jinjin Gu, Tianfan Xue, and Chao Dong. Teaching Large Language Models to Regress Accurate Image Quality Scores using Score Distribution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14483–14494, 2025. URL https://arxiv.org/abs/2501.11561.

Lvmin Zhang, Shengqu Cai, Muyang Li, Gordon Wetzstein, and Maneesh Agrawala. Frame Context Packing and Drift Prevention in Next-Frame-Prediction Video Diffusion Models. In *NeurIPS*, 2025. URL https://arxiv.org/abs/2504.12626.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. In *NeurIPS*, volume 36, 2023. URL https://arxiv.org/abs/2306.14048.

Min Zhao, Hongzhou Zhu, Bokai Yan, Zihan Zhou, Yimin Chen, Wenqiang Sun, Kaiwen Zheng, Guande He, Xiao Yang, Chongxuan Li, Fan Bao, and Jun Zhu. minWM: A Full-Stack Open-Source Framework for Real-Time Interactive Video World Models. *arXiv preprint arXiv:2605.30263*, 2026. URL https://arxiv.org/abs/2605.30263v1.

Xiangqing Zheng, Chengyue Wu, Kehai Chen, and Min Zhang. LoCoT2V-Bench: Benchmarking Long-Form and Complex Text-to-Video Generation. In *ICML*, 2026. URL https://arxiv.org/abs/2510.26412.

Junhao Zhuang, Shiyi Zhang, Yuxuan Bian, Yaowei Li, Yawen Luo, Yijun Liu, Weiyang Jin, Songchun Zhang, Xianglong He, Xuying Zhang, Haoran Li, Haoyang Huang, Zeyue Xue, and Nan Duan. Self Gradient Forcing: Native Long Video Extrapolation. *arXiv preprint arXiv:2607.20368*, 2026. URL https://arxiv.org/abs/2607.20368.

## AI USE STATEMENT

Generative AI assisted language polishing, literature retrieval and organization, research ideation, mathematical checking, drafting paper sections, and LaTeX and figure formatting. The authors retain responsibility for verifying AI-assisted text, citations, mathematical claims and numerical evidence, and for the final scientific content and submission.

## A COMPLETE ALGORITHM AND NOTATION

Algorithm 1 expands (1), (2) and (4) into the complete streaming procedure. The sampling loop and the retention rule are those of the frozen host; ARROW changes the input, the mask and the stored rows of the writing loop. Hosts with several guidance branches apply the writing loop to each branch.

**Masks and positions.** For $r = 2$, $\mathbf{A}_2 = \left(\begin{smallmatrix} 0 & -\infty \\ 0 & 0 \end{smallmatrix}\right)$ at block level, combined with the host's within-block mask; for $r = 1$, $\mathbf{A}_1$ is the native within-block mask. Both blocks keep the temporal positions that the host assigns to them, so the K/V of block $t$ occupy their native position. $\mathcal{P}_\ell$ includes the host's K/V normalization and exactly the positional rotation that its storage convention applies at write time. Text, reference-image and control streams keep their native roles in $B_\ell$.

**State after each write.** After write $t$, the buffer holds $x_t$ and every newly stored layer contains only the rows of block $t$. The stored entry of block $t-1$ keeps its values and positions. The recomputed K/V of block $t-1$ and all hidden states are released once $C_t$ owns its storage. Between blocks, a streaming implementation therefore keeps the native cache, one clean latent block and the decoder state.

**Starting from a given prefix.** For a clean prefix $x_{1:t_0}$, run the writing and retention steps on the prefix blocks in order, set $b = x_{t_0}$ and start sampling at $t_0 + 1$; Figure 1 is generated this way. A continuation that is given an initial cache uses that cache and the last clean block of the prefix as buffer.

**Algorithm 1** ARROW inference: long reading during denoising, local writing of each entry

**Input:** frozen model $\theta$ with $L$ layers; text embedding $c$; number of blocks $T$; tokens per block $N$; sampler times $\tau_0 > \cdots > \tau_S$; clean time $\tau_c$; retention rule $\Pi$; noise $\{\epsilon_t, \eta_{t,0:S-1}\}_{t=1}^{T}$
**Output:** decoded video, streamed block by block

$\mathcal{M}_0 \leftarrow \varnothing, \quad b \leftarrow \varnothing$ ▷ *empty cache and one-block clean buffer*
**for** $t = 1, \ldots, T$ **do**
  $z \leftarrow \epsilon_t$
  **for** $k = 0, \ldots, S-1$ **do** ▷ *denoising: native reader*
    $z \leftarrow \mathrm{Step}_\theta\left(z, \tau_k, \tau_{k+1};\, c, \mathcal{M}_{t-1}, \eta_{t,k}\right)$ ▷ *reads all retained entries*
  **end for**
  $x_t \leftarrow z$ ▷ *completed clean block*
  **if** $t = 1$ **then** $q \leftarrow x_1,\ r \leftarrow 1$ **else** $q \leftarrow [b, x_t],\ r \leftarrow 2$ **end if**
  $\mathcal{I} \leftarrow \{(r-1)N + 1, \ldots, rN\}$ ▷ *rows of block $t$ in $q$*
  $h^1 \leftarrow \mathrm{Embed}_\theta(q;\, c, \tau_c), \quad C_t \leftarrow \varnothing$
  **for** $\ell = 1, \ldots, L$ **do** ▷ *writing: pair pass*
    $(\widetilde{K}^\ell, \widetilde{V}^\ell) \leftarrow \mathcal{P}_\ell(h^\ell)$
    $C_t^\ell \leftarrow (\widetilde{K}^\ell[\mathcal{I}], \widetilde{V}^\ell[\mathcal{I}])$ ▷ *keep rows of block $t$*
    $h^{\ell+1} \leftarrow B_\ell(h^\ell;\, \varnothing, \mathbf{A}_r)$ ▷ *block-causal mask, no old K/V*
  **end for**
  $\mathcal{M}_t \leftarrow \Pi(\mathcal{M}_{t-1}, C_t)$ ▷ *store the current block only*
  $b \leftarrow x_t$; emit $\mathrm{Decode}(x_t)$; release $q$, $h^{1:L+1}$, $\widetilde{K}^{1:L}$, $\widetilde{V}^{1:L}$
**end for**

### A.1 Notation

| Symbol | Meaning |
|---|---|
| $t$, $T$ | Block index and number of blocks in the video. |
| $\ell$, $L$ | Layer index and number of layers that store K/V. |
| $N$ | Video tokens per block. |
| $x_t$ | Clean latent of block $t$; $[x_{t-1}, x_t]$ concatenates two blocks along time. |
| $C_t = \{(K_t^\ell, V_t^\ell)\}_{\ell=1}^{L}$ | Cache entry of block $t$. |
| $\mathcal{M}_t$, $\mathcal{M}_t^\ell$ | Retained cache after block $t$ and its layer-$\ell$ part. |
| $W$ | Number of blocks kept by first-in-first-out retention. |
| $G_\theta$, $U_\theta$, $\Pi$ | Denoising pass (reader), writing pass (writer) and retention rule. |
| $h_t^\ell$, $\mathcal{P}_\ell$, $B_\ell$ | Hidden state at the input of layer $\ell$, K/V projection and layer update. |
| $\cdot\vert_{x_t}$ | Restriction to the $N$ rows of block $t$ at every layer. |
| $\mathbf{A}_r$ | Block-causal mask of the writing pass over $r \in \{1, 2\}$ clean blocks. |
| $\epsilon_t$, $\tau_k$, $S$ | Initial noise of block $t$, sampler times and number of sampling steps. |
| $\tau_c$ | Native clean time of the writing pass (0 for Wan and Hunyuan, 0.9999 for MAGI). |
| $Y$, $X$, $X^-$, $Z$ | Next-block target, current block, previous block and other reader inputs (Appendix B.2). |
| $D_j$, $\rho_\theta$, $\mathcal{L}_j$ | Candidate entry, frozen reader and its prediction risk with $j$ clean blocks. |
| $\mathcal{F}_j$, $m_j$, $R_j$, $e_j$ | Information from $j$ clean blocks, best predictor, its risk and the reader's excess risk. |

## B Proofs and Analysis

### B.1 Lifetime of Cache Differences under Fixed Video

Time $t$ counts writes. The two executions in Theorem 1 share block indices, shapes, positions, clean latents, clean buffers and external inputs, and their initial K/V values differ. Computation is deterministic, and once the cache is full every layer retains exactly the latest $W$ entries. These

conditions leave the old K/V as the only source of difference, so the analysis follows how that difference moves through the writing pass.

In the writing pass of (2), the input of layer 1 is computed from the current latent and the external inputs. Each layer projects its stored K/V from its input state before it attends to the cache, and passes the attended state to the next layer. For a host with several streams, the input state includes all streams of the current pass. This order gives the following lemma.

**Lemma 1** (Stored K/V depend only on lower cache layers)**.** *Under these conditions, a native entry at layer $\ell$ has the form*

$$C_t^\ell = f_{\ell,t}\big(x_t,\, \mathcal{M}_{t-1}^1, \ldots, \mathcal{M}_{t-1}^{\ell-1}\big). \tag{5}$$

*In particular, $C_t^1$ does not depend on old K/V.*

*Proof.* The input $h_t^1$ does not depend on the cache, so neither does $\mathcal{P}_1(h_t^1)$. Suppose $h_t^\ell$ depends only on current inputs and on cache layers below $\ell$. Its projection has the same dependence. The update $B_\ell$ additionally reads $\mathcal{M}_{t-1}^\ell$, so $h_t^{\ell+1}$ depends on cache layers up to $\ell$. Normalization, residual connections and fixed positional transforms preserve this order. □

*Proof of Theorem 1.* For a writer without old K/V, each new entry is a deterministic function of shared clean latents and inputs, so new entries agree from the first write, and after $W$ writes the cache contains only such entries. For the native writer, Lemma 1 gives equal new layer-1 entries from write 1, hence equal layer-1 caches from write $W$. Suppose every cache layer below $\ell$ agrees from write $(\ell-1)W$. At every later write the lower-layer inputs of (5) agree, so the new layer-$\ell$ entries agree, and one further window removes the remaining unequal layer-$\ell$ entries. Layer $\ell$ therefore agrees from write $\ell W$, and the whole cache from write $LW$. □

The argument allows clean inputs and positions that vary over time, as long as the two executions share them. A current-only native writer, such as that of MAGI-1, also produces equal new entries immediately and has the same $W$ bound as ARROW.

**The layered bound is attained.** Consider scalar entries with

$$C_t^1 = 0, \qquad C_t^\ell = \alpha\, C_{t-W}^{\ell-1} \quad (\ell \geq 2,\ \alpha \neq 0), \tag{6}$$

so that each layer copies the oldest retained entry of the layer below. An initial difference $\delta$ in $C_0^1$ reaches $C_W^2$ as $\alpha\delta$, $C_{2W}^3$ as $\alpha^2\delta$ and finally $C_{(L-1)W}^L$ as $\alpha^{L-1}\delta$. This last entry is retained through write $LW-1$ and evicted at write $LW$. If the lowest initially differing layer is $\ell_0$, the same argument gives the bound $(L-\ell_0+1)W$. The bound is finite because a Transformer writer copies strictly upward in depth: a recurrence within one layer, $C_t = x_t + \gamma C_{t-1}$ with $0 < |\gamma| < 1$, keeps $\gamma^t\delta$ for every finite $t$.

**Retention rules with their own state.** The theorem assumes first-in-first-out retention, which does not look at entry values. Other rules carry state of their own. A persistent sink keeps an initial difference indefinitely, and recursive merging copies it into a summary entry. Adaptive selection can also carry it: if every new entry contains $(-1,+1)$ and a one-entry selector keeps the value whose sign matches the previous cache, the initial sign survives all replacements even with a writer that ignores old K/V. The hosts in Section 5.6 keep their own sinks and selection, and ARROW removes the writer's contribution in them.

**Dependence through the generated video.** In free generation, an initial perturbation $\lambda$ also changes the clean blocks. With external inputs fixed, the ARROW entry $C_t = U_\theta([x_{t-1}, x_t]; \varnothing)|_{x_t} =: \Phi_\theta(x_{t-1}, x_t)$ obeys

$$\frac{\mathrm{d}C_t}{\mathrm{d}\lambda} = \frac{\partial \Phi_\theta}{\partial x_{t-1}} \frac{\mathrm{d}x_{t-1}}{\mathrm{d}\lambda} + \frac{\partial \Phi_\theta}{\partial x_t} \frac{\mathrm{d}x_t}{\mathrm{d}\lambda}. \tag{7}$$

This is the first term of (3): history reaches the entry through the video that the reader generated. For example, a reader with $x_t = \beta x_{t-1}$ carries a perturbation proportional to $\beta^t$ into later blocks. The theorem fixes this route by replaying the same video, and the free-generation experiments measure its effect on the output.

### B.2 Prediction Risk of a Cache Entry

The prediction test in Section 5.5 replaces only the entry of block $t$ and measures the frozen network's error for the next block. This subsection states what that error difference measures. Let $Y$ be the denoising target of block $t+1$, $X = x_t$ the current block, $X^- = x_{t-1}$ the previous block and $Z$ all other reader inputs, including the rest of the cache and the noisy next block. All variables share one probability space and have finite second moments; for an $n$-coordinate vector, $\|v\|^2 = n^{-1}\sum_{i=1}^{n} v_i^2$. Define $\mathcal{F}_1 = \sigma(Z, X) \subseteq \mathcal{F}_2 = \sigma(Z, X^-, X)$, the best predictors $m_j = \mathbb{E}[Y \mid \mathcal{F}_j]$ and their risks $R_j = \mathbb{E}\|Y - m_j\|^2$. For the deterministic entries $D_1 = \phi_1(X)$ and $D_2 = \phi_2(X^-, X)$, the frozen reader $\rho_\theta$ has risk $\mathcal{L}_j = \mathbb{E}\|Y - \rho_\theta(Z, D_j)\|^2$ and excess risk $e_j = \mathcal{L}_j - R_j$.

**Proposition 1** (Information and reader error)**.** *With $\Delta_{\text{pred}} = \mathbb{E}\|m_2 - m_1\|^2$, the risk difference of the two entries is*

$$\mathcal{L}_1 - \mathcal{L}_2 = \underbrace{\Delta_{\text{pred}}}_{\text{information in the previous block}} + \underbrace{e_1 - e_2}_{\text{change in encoding and reader error}}, \qquad e_j \geq 0. \tag{8}$$

The pair entry lowers the error when the information in the previous block exceeds any increase in reader error. It can also lower the error when $Z$ already contains the previous block, by encoding it in a form that the frozen reader uses better. The measured MSE difference is the sum in (8).

The information fields of the entries satisfy

$$\mathcal{D}_j = \sigma(Z, D_j) \subseteq \mathcal{F}_j, \qquad \mathcal{F}_1 = \sigma(Z, X) \subseteq \mathcal{F}_2 = \sigma(Z, X^-, X). \tag{9}$$

The inputs may be dependent; in particular, $Z$ may encode the previous block. The reader output $\rho_\theta(Z, D_j)$ is $\mathcal{D}_j$-measurable and hence $\mathcal{F}_j$-measurable.

**Lemma 2** (Squared-risk identity)**.** *For a sigma-field $\mathcal{H}$, let $m_{\mathcal{H}} = \mathbb{E}[Y \mid \mathcal{H}]$. Every square-integrable, $\mathcal{H}$-measurable predictor $g$ satisfies*

$$\mathbb{E}\|Y - g\|^2 = \mathbb{E}\|Y - m_{\mathcal{H}}\|^2 + \mathbb{E}\|m_{\mathcal{H}} - g\|^2. \tag{10}$$

*Proof.* Write $Y - g = (Y - m_{\mathcal{H}}) + (m_{\mathcal{H}} - g)$. The cross term has zero expectation because $\mathbb{E}[Y - m_{\mathcal{H}} \mid \mathcal{H}] = 0$ and $m_{\mathcal{H}} - g$ is $\mathcal{H}$-measurable. □

*Proof of Proposition 1.* Since $\mathcal{F}_1 \subseteq \mathcal{F}_2$, Lemma 2 with $\mathcal{H} = \mathcal{F}_2$ and $g = m_1$ gives $R_1 - R_2 = \mathbb{E}\|m_2 - m_1\|^2 = \Delta_{\text{pred}}$. With $\mathcal{H} = \mathcal{F}_j$ and $g = \rho_\theta(Z, D_j)$ it gives $e_j = \mathbb{E}\|m_j - \rho_\theta(Z, D_j)\|^2 \geq 0$. Subtracting $\mathcal{L}_j = R_j + e_j$ for $j = 1, 2$ yields (8). □

**Encoding loss and reader error.** Let $\widehat{m}_j = \mathbb{E}[Y \mid \mathcal{D}_j]$. Two applications of Lemma 2 split the excess risk into the information lost by encoding clean latents into K/V, $\kappa_j = \mathbb{E}\|m_j - \widehat{m}_j\|^2$, and the error of the actual reader given that entry, $\omega_j = \mathbb{E}\|\widehat{m}_j - \rho_\theta(Z, D_j)\|^2$:

$$e_j = \kappa_j + \omega_j, \qquad \mathcal{L}_1 - \mathcal{L}_2 = \Delta_{\text{pred}} + (\kappa_1 - \kappa_2) + (\omega_1 - \omega_2). \tag{11}$$

The clean-latent fields are nested, while $\mathcal{D}_1$ and $\mathcal{D}_2$ need not be, so the last two differences can take either sign. Two scalar examples show both cases. First, let $X^-$ be uniform on $\{-1, +1\}$, $X = 0$, $Y = Z = X^-$, let the writer average its inputs and let the reader be $\rho(Z, D) = 2D$. Then $D_1 = 0$, $D_2 = X^-/2$ and $(\mathcal{L}_1, \mathcal{L}_2) = (1, 0)$ although $\Delta_{\text{pred}} = 0$: the new entry makes information already in $Z$ usable by this reader. Second, let $X$ be uniform on $\{-1, +1\}$, $X^- = -X$, $Y = X$, $Z$ constant, the same writer and $\rho(Z, D) = D$. Then $(D_1, D_2) = (X, 0)$ and $(\mathcal{L}_1, \mathcal{L}_2) = (0, 1)$: the extra block changes the encoding and loses the target. The prediction test therefore measures the net effect of the pair on the frozen reader.

**Longer windows.** On common positions, let $\mathcal{F}_r = \sigma(Z, x_{t-r+1}, \ldots, x_t)$, $m_r = \mathbb{E}[Y \mid \mathcal{F}_r]$ and $\Delta_r = \mathbb{E}\|m_{r+1} - m_r\|^2$. The same proof gives

$$R_r - R_{r+1} = \Delta_r \geq 0, \qquad \mathcal{L}_r - \mathcal{L}_{r+1} = \Delta_r + e_r - e_{r+1}, \tag{12}$$

and the ideal gains add up as $R_1 - R_r = \sum_{j=1}^{r-1} \Delta_j$. Their sizes depend on the data. For independent scalar-sign blocks and the target $Y = x_{t-2}$, the second block contributes nothing and the third

determines the target. In video, the previous block contains the transition into block $t$, and Tables 3 and 6 show that a third block adds little.

**A transition example.** Let $Y = 2X - X^- + \zeta$ be the next position of a point moving at constant speed, with $Z$ constant and independent zero-mean noise $\zeta$. Then $m_2 = 2X - X^-$ and $m_1 = 2X - \mathbb{E}[X^- \mid X]$, so

$$\Delta_{\text{pred}} = \mathbb{E}\big[\,\mathrm{Var}(X^- \mid X)\big]. \tag{13}$$

The gain is the uncertainty about where the point came from, given where it is. A video block contains several frames and already shows some motion internally; the previous block adds the frames before the boundary, where the current block leaves the most uncertainty, such as at the onset of a motion or a change in visibility.

### B.3 Storage and Computation

Relative to a current-only pass, the pair doubles the token-linear work of projections and feed-forward layers. At block level, its mask allows $N^2$ interactions within the previous block and $2N^2$ for the current block, $3N^2$ in total, while a one-block writer that reads $W$ cached blocks allows $(W+1)N^2$. External streams and kernel layouts add their own work, and a dense kernel computes the full $2N \times 2N$ score matrix before masking. With fixed cache size, block shape, model and sampler, memory and per-block work stay constant along the video, and total work grows linearly with the number of blocks.

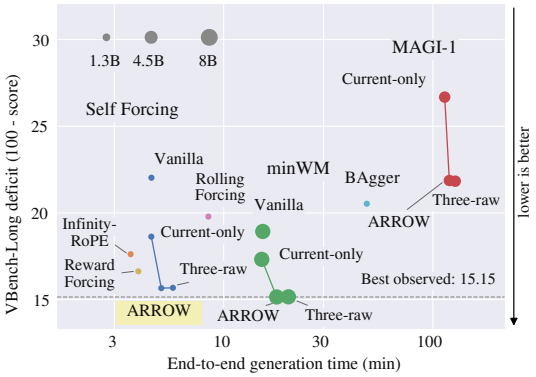



Figure 9: Quality versus cost. VBench-Long deficit (100 minus the Total score) against generation time (log scale); lower left is better. Connected points are writer variants of one model (Vanilla: native; Three-raw: three blocks); marker size denotes model size.

Figure 9 places the writer variants of Table 3 and the external systems of Table 1 on one quality–cost plane. Within each model, the pair moves the operating point well down at a modest increase in time, and the three-block writer moves it right at almost no gain in quality. Distances between models also reflect resolution and sampler differences, so the plot compares writers within one model.

For $N$ video tokens per block, $W$ retained blocks, $n_{\text{B}}$ bytes per element, $H_\ell$ K/V heads and head dimension $d_\ell$, the cache occupies

$$\mathrm{Mem}_{\text{KV}} = 2\, n_{\text{B}}\, W N \sum_{\ell=1}^{L} H_\ell d_\ell \tag{14}$$

bytes. ARROW keeps this size and adds one clean latent block of $n_{\text{lat}}$ elements, or $n_{\text{B}} n_{\text{lat}}$ bytes. Temporary activations of the writing pass, attention workspaces, weights and decoder state add to peak memory separately.

**Measurement.** Generation time starts from prompt processing with weights resident and ends after the final output is written and synchronized. It includes text encoding, every sampling step, the writing pass, VAE decoding, output handling and the final flush. Per-video allocated and reserved peaks are recorded separately, and Table 1 reports the mean allocated peak. Weights, fixed conditioning, host indices, VAE state and required GPU or CPU buffers count as persistent state; memory shared by two views counts once, and CPU offload counts as state. Outputs are streamed to storage. The BF16 clean buffers occupy 599,040 bytes for Wan, 1,555,200 for MAGI and 399,360 for Hunyuan; the peak-memory increase in Table 1 is temporary memory of the writing pass. Initial, per-block and final contributions add up to the full generation time.

# C Experimental Protocols

## C.1 Frozen Models and Native Interfaces

All matched runs use BF16, batch size one and a single A100 GPU. Within each pair, initialization, precision-sensitive operations, attention kernels, text processing and VAE handling are identical. ARROW uses each frozen model with its native sampler and replaces only the writing pass, as in Algorithm 1.

**Wan.** The Self Forcing DMD EMA checkpoint uses the Wan2.1-T2V-1.3B backbone with 30 layers and 12 K/V heads of dimension 128. A clean block of shape $16 \times 3 \times 60 \times 104$ holds 4,680 video tokens and decodes to 480×832 frames at 16 fps. The four normalized sampler times are $(1, 0.9375, 5/6, 0.625)$, and the writing pass uses time zero. A 21-latent-frame allocation holds seven block slots; attention reserves the slot of the current block and reads up to 18 cached latent frames, so $W = 6$. The weights are `self_forcing_dmd.pt` from `gdhe17/Self-Forcing`, revision `2f8b779212da`, with code revision `33593df3e81f`.

**MAGI.** The distilled MAGI-1-4.5B checkpoint uses 34 layers, 24 query heads and eight K/V heads of dimension 128. A block of shape $16 \times 6 \times 90 \times 90$ holds 12,150 tokens and decodes to 720×720 frames at 24 fps. Inference uses 16 steps, serial window size one, one guidance branch and clean time 0.9999. The native writer is current-only. Five completed blocks are retained; the native denoising-stage ranges $(5, 4, 3, 2)$ count the current block and therefore expose at most four cached blocks during denoising. FP8 and external feature-reuse accelerators are disabled. The model, ViTVAE and T5 share `sand-ai/MAGI-1` revision `ca8bdfb28b1d`, with code revision `0fcefdef8ce2`.

**Hunyuan.** The minWM HY15/TI2V/DMD 8B checkpoint has 54 dual-stream layers and 16 K/V heads of dimension 128. A block of shape $32 \times 4 \times 30 \times 52$ holds 6,240 tokens and decodes to 480×832 frames at 16 fps. Four Euler steps use the Wan sampler times, guidance scale one and clean time zero. Both writers use a 20-latent-frame, five-block first-in-first-out cache ($W = 5$) with native absolute RoPE and the text-to-video branch. The weights are revision `21bd74da43b5` of `MIN-Lab/minWM`, with code revision `75322cc41e1d`.

**Prompts and durations.** Every system receives the complete original prompt; LoCoT2V scene and character annotations are used only by its evaluators. Wan uses a text length of 576 for LoCoT2V and 512 otherwise, and MAGI and Hunyuan keep their native limits of 800 and 1,000. For a requested duration $d_{\text{req}}$, frame rate $\nu$, $f$ latent frames per block and decoder offset $o$, the shortest complete output has

$$T(d_{\text{req}}) = \left\lceil \frac{\lceil \nu d_{\text{req}} \rceil + o}{4f} \right\rceil \text{ blocks}, \qquad F_{\text{out}} = 4fT(d_{\text{req}}) - o \text{ frames}. \tag{15}$$

Wan and Hunyuan have $o = 3$ and MAGI has $o = 0$. Five-second requests yield 81, 120 and 93 frames (5.06, 5.00 and 5.81 seconds) for Wan, MAGI and Hunyuan, and sixty-second requests yield 969, 1,440 and 973 frames (60.56, 60.00 and 60.81 seconds), that is, 81, 60 and 61 blocks. LoCoT2V uses each prompt's reference duration, from 29.75 to 64.24 seconds, with the same rule. Every output keeps its complete final block, so both writers of a pair produce the same number of frames.

## C.2 External Systems

Deep Forcing (Yi et al., 2026) and Infinity-RoPE (Yesiltepe et al., 2026) use the Self Forcing checkpoint of the Wan pair and change retention and temporal addressing. Deep Forcing enables Deep Sink and Participative Compression at its published setting $S/N/R = 10/16/4$ with 21 latent-frame slots; selected tokens keep their original K/V values. Infinity-RoPE uses four sampling steps, a six-latent-frame allocation that includes the current three-frame block, onset 21 and shift five, and keeps its block-relative positions and sink policy. The prompts contain no prompt changes or cuts, so these rows compare complete inference policies on shared weights.

Rolling Forcing (Liu et al., 2026) keeps its five-step rolling process, three recent and three global clean latent frames and a 15-latent-frame noisy window; it was trained with a 27-latent-frame window for 3,000 iterations at batch size eight. LongLive (Yang et al., 2026) uses the checkpoint trained with 60-second rollouts, four steps, nine local latent frames and three sink frames. Self-Forcing++ (Cui et al., 2025) is the main EMA model of the paper, trained with backward-noise DMD, with

four-step inference and 21-latent-frame retention, before optional GRPO. Reward Forcing (Lu et al., 2026) keeps Re-DMD and its EMA sink, four steps, nine local plus three sink latent frames and motion-reward weight $0.5$; it was trained for 600 iterations at batch size 64. BAgger (Po et al., 2026) uses its Round-3 model with 20 steps, guidance scale six and shift eight. Its 21-latent-frame window refreshes K/V from a 12-latent-frame clean window, and both guidance branches count toward state and runtime; initial training and each aggregation round use 16,000 iterations at batch size 96.

The same nine systems are evaluated on VBench-Long, short VBench, LoCoT2V-Bench, T2V-CompBench, VideoPhy-2, VMBench and the Movie Gen preference prompts. Each benchmark keeps its own prompts, metrics, units and duration rule.

### C.3 Benchmarks and Aggregation

**VBench.** The 946 official records contain 944 distinct prompts; repeated text shares its generated video and keeps every official metric membership. Ordinary dimensions use seeds 0–4, and the 75 temporal-flickering prompts add seeds 5–24, giving 6,220 generated videos per duration and system. Each dimension is aggregated over its own prompts before the official normalization. Quality averages seven normalized dimensions, with weight $0.5$ on Dynamic Degree and one on the others; Semantic averages nine dimensions; Total weights Quality and Semantic by four and one. For long videos, the pinned implementation splits each video into clips at source frame rate. A prompt's clip length is the longest duration required by its dimensions, and the final clip borrows preceding frames to remain complete. Subject and background consistency combine within-clip scores with the official interpolated between-clip score for each video before aggregation. The static filter processes all 25 seeds and keeps at most five short videos or five long clips per prompt; excluded items leave both numerator and denominator. All generated candidates count toward resource measurements.

**LoCoT2V-Bench.** All 234 prompts and their published durations are used (Zheng et al., 2026). The evaluator uses DeQA (You et al., 2025) for perceptual quality, Qwen3-VL-8B-Instruct (Bai et al., 2025) for overall and fine-grained alignment and HERD, and the official SAM 3 (Carion et al., 2025), Qwen3-VL-4B-Instruct and FG-CLIP 2 (Xie et al., 2026) pipeline for character consistency. Fine-grained alignment keeps the scene-existence gates and the nested character, attribute and action counts, so a scene or entity earns credit only after passing its gate. HERD keeps its published expectations and two-stage prompts. Warping error is converted to a score by $\exp(-5\,\mathrm{error})$, and temporal quality is $\mathrm{TQ} = (\mathrm{CC} + \mathrm{BC} + \mathrm{WE})/3$. Dynamic quality uses the published standardizer and regression coefficients. Displayed LoCoT2V values are multiplied by 100.

**Composition, physics and motion.** T2V-CompBench (Sun et al., 2025) uses 200 prompts in each of seven categories with the official judge repeats and category-specific scoring; dynamic-attribute binding requires the old attribute at the start, the new attribute at the end and positive intermediate evidence. VideoPhy-2 (Bansal et al., 2026) evaluates 591 test prompts using integer semantic and physical ratings from one to five; joint success requires both ratings to be at least four. VMBench (Ling et al., 2025) uses all 1,050 prompts and its five official scoring pipelines, with expected-class scores for class-based predictions. These suites use five-second requests and three seeds. Scores are multiplied by 100 where the tables state it.

### C.4 Paired Statistics

The sampling unit is the distinct prompt. A bootstrap replicate gives the same prompt weight to every method, seed, reference source and intervention branch attached to that prompt, keeps each metric's numerator and denominator, recomputes the official aggregate and then forms the paired contrast. VBench dimensions keep their own prompt sets, compositional categories are resampled separately, and benchmarks are never pooled. We use 10,000 resamples with random seed 20260909 and report the 2.5th and 97.5th percentiles. Frames, blocks, characters and judge repeats stay clustered within their prompt.

For the prediction test, each video contributes its squared-error sum divided by its number of target elements; seeds are weighted equally within a prompt and prompts equally across the benchmark, so long LoCoT2V videos and the extra VBench flicker seeds do not dominate. Resource means use every generated video, including those removed by the static filter. Tables report means, and the text reports paired intervals.

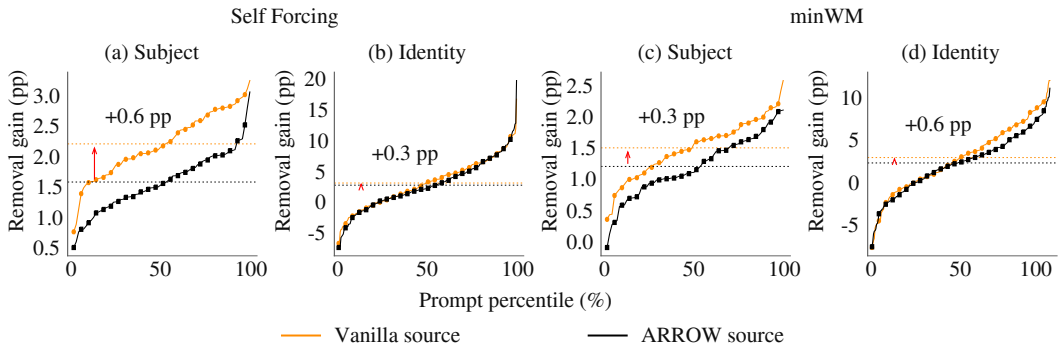



Figure 10: Per-prompt gains from removing old K/V in the replay intervention, measured after all intervention entries are evicted. Panels show subject consistency and identity (LoCoT2V character consistency) for Wan (Self Forcing) and Hunyuan (minWM), with prompts sorted within each reference source (Vanilla: native). Dotted lines mark mean gains; arrows mark the difference between sources.

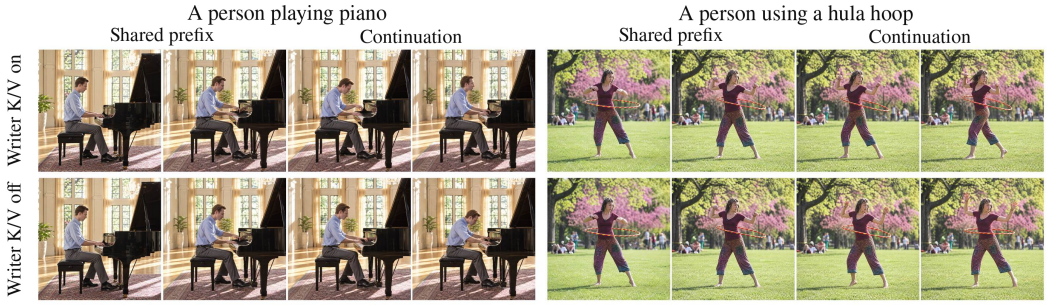



Figure 11: Paired continuations from the replay intervention on Wan, with old K/V in the writer during the intervention (Writer K/V on) and without (Writer K/V off). Two shared-prefix columns precede two continuation columns for each prompt. Compare the pianist's hands and posture and the dancer's arms and hoop at matching columns.

### C.5 Controlled Comparisons

The three controlled comparisons of Section 5 each hold everything fixed except one quantity. The replay intervention fixes the clean video and varies whether the writer reads old K/V. The prediction test fixes the video, the cache and the next noisy block and varies only the entry of one block. The reader intervention fixes the writer and varies how much retained history the reader may attend to. Each therefore isolates one element of ARROW.

**Replay intervention.** For each reference video, the shared initial cache is its state at the first write that fills the window; call this write $t_s$. Both branches copy this state together with all external, positional and random-number state. Old K/V are present or absent only during writes $t_s + 1$ to $t_s + W$, and in this period both branches write each block from its own clean latent without the previous block, so the branches differ only in the old K/V. Both then apply the native writer to identical reference latents through write $t_s + 2W$, and free generation runs from there to the original end of the video. The evaluated video includes the common prefix and the replayed part. For native reference videos, the branch with old K/V reproduces the original native run, whose result is reused.

Figure 10 breaks the mean effects of Table 2 down by prompt. Subject-consistency gains are positive for nearly all prompts in both models, so the harm of copied K/V is systematic and not driven by a few videos. Identity gains have a wider spread and are negative for up to a quarter of the prompts, while their means stay positive; the identity score is computed from characters detected and matched in each clip (Appendix C.3), so it varies more from prompt to prompt. Native-source curves (orange) lie above ARROW-source curves (black) at most percentiles, as discussed in Section 5.3.

Figure 11 shows two such pairs. The prefix columns come from the replayed reference video, and the continuation columns differ only through entries written after the intervention; the differences are local, in hand and arm pose and in the position of the hoop, consistent with the per-prompt subject-consistency gains of 0.5–3 points on Wan in Figure 10.

**Prediction test.** A reference video has $T$ blocks. The current-only versus pair comparison uses $t = 2, \ldots, T-1$, so that both the previous block and the next-block target exist; the three-block comparison uses the common positions $t = 3, \ldots, T-1$. Each candidate replaces only the entry of block $t$. The other entries and their positions, the noisy block $t+1$, the external inputs and the network weights are identical. A sampler position is drawn uniformly, and all candidates share its noise $\xi$. For Wan and Hunyuan, the noisy input is $(1-u)x_{t+1} + u\xi$ at the four sampler times above, with target $\xi - x_{t+1}$. For MAGI, with $v_k = k/16$ and $u_k = v_k^2/(3-2v_k^2)$ for $k = 0, \ldots, 15$, the noisy input is $u_k x_{t+1} + (1-u_k)\xi$, the target is $x_{t+1} - \xi$, and the cached range follows $\lfloor k/4 \rfloor$ as in the native schedule. The targets differ between models, so MSE values are compared within a model.

**Reader intervention.** Each Deep Forcing or Infinity-RoPE run first applies its own retention and position rules. After the cache is first full, the adjacent reader removes during denoising every cached video entry whose source block is older than the previous block, including sink entries. The native writer keeps its native input and ARROW keeps the pair input. Deep Forcing's content-dependent selection follows each branch's own trajectory, so the reader contrast includes its effect on later selection. Writer effects, reader effects and their difference are formed within a host with shared bootstrap weights.

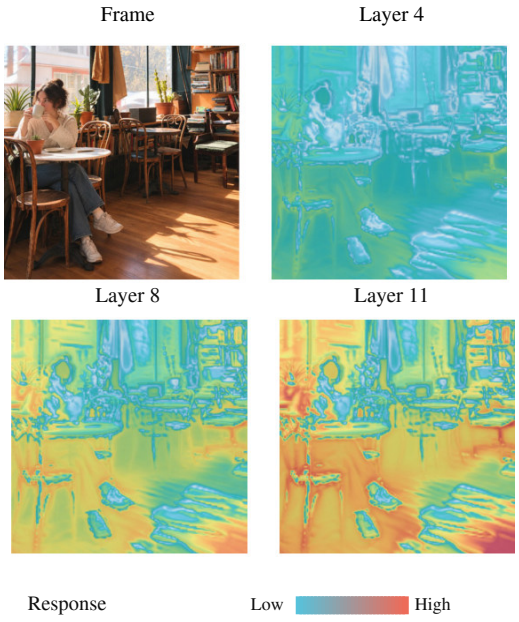



Figure 12: Hidden-state response to readable older entries at Layers 4, 8 and 11; shared crop and scale.

Figure 12 shows where the network responds to older entries before the two readers produce different videos. The input frame is fixed, so the maps isolate the effect of readable history on the hidden states at three depths. The response grows with depth: it is weak everywhere at Layer 4, and at Layers 8 and 11 it is strongest over large scene regions such as the floor, the walls and the window, and weakest along the outlines of the person, chairs and tables. Older entries therefore act mainly on the layout and surfaces of the scene, which agrees with the background-structure example in Figure 8. Later video quality depends on many such updates, and Table 5 measures their effect on complete videos.

# D Additional Results

## D.1 Writer Variants and the Stored Predecessor Entry

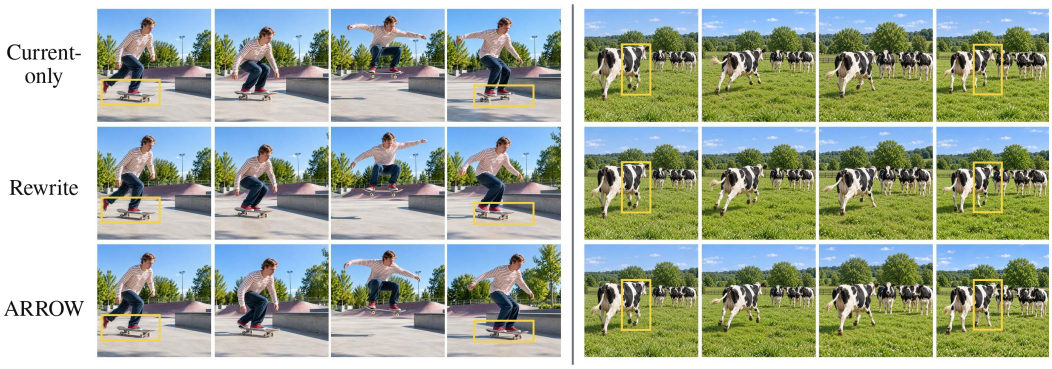



Figure 13: Writer variants on Wan: Current-only, Rewrite (pair writing that also overwrites the previous block's entry) and ARROW, with four frames per case. Yellow boxes mark the board and feet and the front cow's legs at the first and last displayed frames.

Figure 13 shows the writer ablation of Table 3 on two videos. All three writers encode the first block from its own latent, so the first displayed frames coincide, and the rows separate only as later entries differ. Rewrite and ARROW run the same pair pass at every write and differ only in whether the stored entry of the previous block is replaced, so these two rows isolate the storage rule; Current-only differs from ARROW in the input of every write. The skater performs the same jump in every row and the cow runs past the same herd, so the comparison concerns local structure: the boxes mark the board and feet and the front cow's legs and coat pattern at the first and last displayed frames. Table 3 aggregates these local differences over all prompts.

The storage rule is visible in the entries themselves (Figure 4): the stored entry of a block was computed with the preceding frames in view, whereas the recomputed one encodes the block as the first element of a pass, and the two differ over the whole frame. Each entry is therefore written once, with its own incoming transition, and kept; the previous block's clean latent serves only as input to the next write. Every stored entry is then a function of exactly two consecutive clean blocks, the property used in the proof of Theorem 1.

### D.2 Long-Horizon Appearance

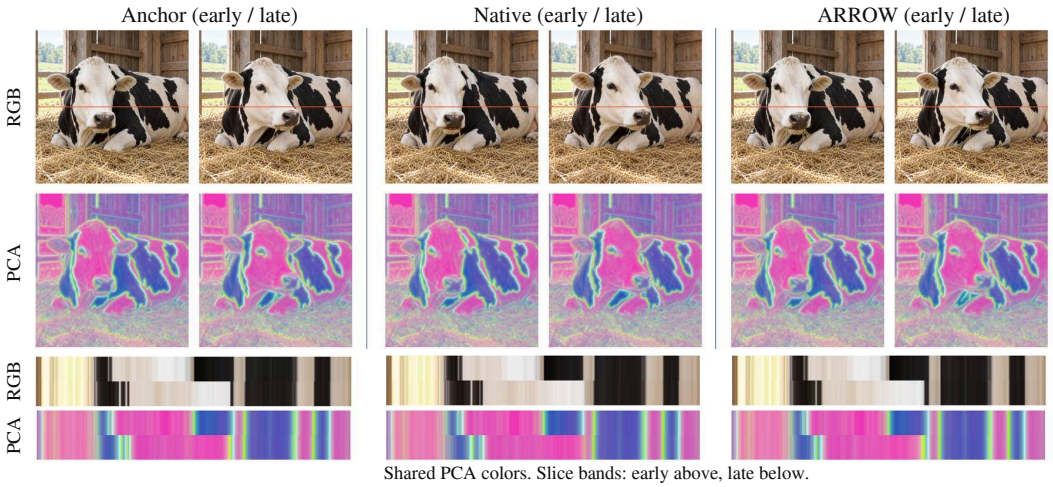



Figure 14: Early and late frames of the cow scene (RGB) and their local descriptors in one shared PCA basis (PCA); bottom bands follow the red scanline, early above late.

Figure 14 examines appearance change in one video at the level of local descriptors. Anchor, Native and ARROW each contribute an early and a late frame. Descriptors combine RGB, local color and gradients, and all six frames share one principal-component basis, so equal hues denote similar local appearance. The scanline bands reduce each frame to one row: a smooth displacement shifts an edge along the band, whereas a changed coat marking or wall alters the colors themselves. Reading the head, the coat markings and the barn structure in this way separates movement of the same content from a change of content, the distinction that the between-clip scores in Figure 6 summarize over many prompts.

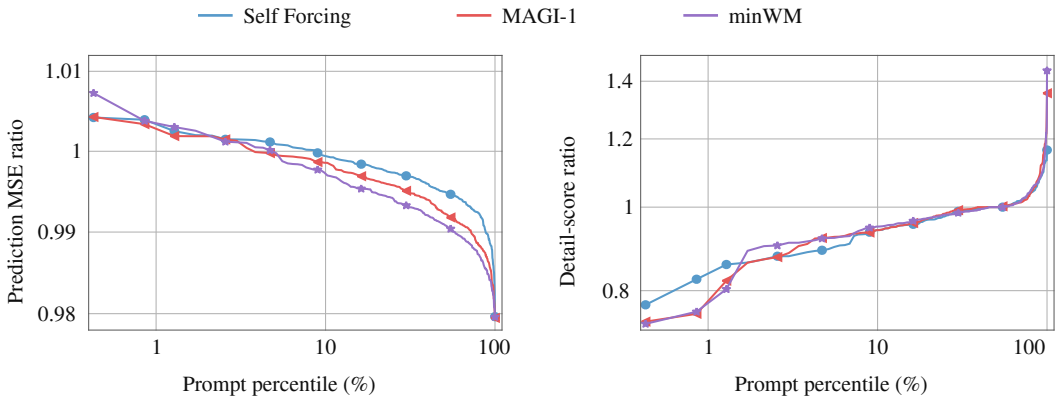
### D.3 Spatial Prediction and a Third Block

Section 5.4 found that a third clean block barely changes video quality. This subsection checks the same question at the level of the frozen reader's prediction, per position and per prompt. Figure 15 compares the returns of the third block per prompt. The MSE ratio of three blocks to the pair lies between 0.98 and 1.01 for all prompts in all three models, so the third block never changes the prediction much. The ratio of detail scores in the generated videos is below one for about 60% of prompts and above one for the rest, mostly within 0.9 to 1.1: the third block adds detail to some videos and removes it from others, in line with the unchanged average quality in Table 3. Since it also adds 7–14% generation time, the pair is the better operating point.

Table 6: **Shared-reader prediction with three windows.** Mean MSE ($10^{-2}$; lower is better) on $t = 3, \ldots, T-1$. Native/ARROW identify reference trajectories. Targets and other reader inputs are shared. Bold marks within-host optima.

| Writer | VBench Long ↓ | | LoCo ↓ | |
|---|---|---|---|---|
| | Native | ARROW | Native | ARROW |
| *Wan2.1-1.3B (Huang et al., 2025; Wan Team et al., 2025)* | | | | |
| 1 block | 19.01 | 16.60 | 23.64 | 20.71 |
| 2 blocks | 16.71 | 15.10 | 20.25 | 18.31 |
| 3 blocks | **16.50** | **15.05** | **20.05** | **18.22** |
| *MAGI-1-4.5B (Sand.ai et al., 2025)* | | | | |
| 1 block | 28.29 | 26.70 | 32.20 | 29.36 |
| 2 blocks | 25.10 | 24.50 | 28.29 | 26.46 |
| 3 blocks | **24.89** | **24.40** | **27.99** | **26.26** |
| *HunyuanVideo1.5-8B (Wu et al., 2025; Zhao et al., 2026)* | | | | |
| 1 block | 11.70 | 10.69 | 14.34 | 12.79 |
| 2 blocks | 10.29 | 9.69 | 12.33 | 11.19 |
| 3 blocks | **10.23** | **9.67** | **12.22** | **11.09** |

Table 6 repeats the prediction test on the positions where three blocks are available. For Wan on native VBench videos, the pair lowers MSE from 19.01 to 16.71 and three blocks lower it to 16.50. The ordering holds in every column: the first reduction is 1.00–3.91 and the second 0.02–0.30, at most about one tenth of the first. This ratio is similar in the three models, although their targets and scales differ, so the diminishing return is a property of the video data. By (12), the ideal gain of each added block is the information it adds about the target; the small second step shows that most of this information lies in the block adjacent to the boundary.




(a) Prediction error (b) Detail score

Figure 15: Returns from a third clean block, per prompt. (a) MSE ratio of three blocks to ARROW on matching targets and positions (below one favors three blocks). (b) Ratio of prompt-paired detail scores over three generations (above one favors three blocks).

### D.4 Short-Horizon Capabilities

Short requests test whether ARROW changes capabilities that do not depend on many cache writes. A five-second video involves five to seven writes, so differences between writers have few steps over which to compound. Table 7 confirms this for short VBench: the paired quality difference is 0.03 points for Wan (95% CI $-0.04$ to $0.11$) and 0.01 for MAGI and for Hunyuan (each $-0.06$ to $0.09$). Differences are computed before display rounding. The comparison with Table 1 is informative for motion. Native Wan's Dynamic Degree falls from 69.7 for five-second videos to 44.0 for one-minute videos, whereas ARROW's changes from 70.0 to 66.8; on MAGI and Hunyuan, ARROW's long-video Dynamic Degree exceeds its short-video value. Motion that the native writer loses over many writes is thus retained with ARROW.

Among the external systems, short-video leaders differ by dimension: Deep Forcing has the highest quality (85.24), Reward Forcing the highest semantic score (81.39), LongLive the highest subject, background, flickering, smoothness and aesthetic scores, Rolling Forcing the highest imaging quality and BAgger the highest Dynamic Degree (78.9). The matched Wan pair differs by at most 0.3 points on every dimension, so ARROW keeps the short-video profile of its base model.

Table 7: **Short-horizon VBench with matched and external systems.** Official 100-point aggregates for five-second requests completed at native-block boundaries; means only. Native–ARROW comparisons share generation settings within each architecture; external Wan systems retain their settings. Quality/Semantic, Sub./Bg., and DD/MS follow Table 1; Flick.: temporal flickering; Aes./Img.: aesthetic/imaging quality; Overall: overall consistency. All columns increase with quality. Bold marks architecture-wise optima. The ordinary generation seeds are 0–4; flickering retains its separate official filtered population.

| Method | VBench (five-second requests) ↑ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Qual. | Sem. | Sub. | Bg. | Flick. | MS | DD | Aes. | Img. | Overall |
| *Wan2.1-1.3B / Self Forcing (Huang et al., 2025; Wan Team et al., 2025)* | | | | | | | | | | |
| Native | 84.87 | 80.33 | 95.13 | 96.13 | 99.00 | 98.25 | 69.7 | 65.85 | 69.71 | 26.99 |
| Deep Forcing (Yi et al., 2026) | **85.24** | 80.45 | 95.24 | 96.35 | 98.92 | 98.17 | 75.0 | 65.46 | 69.94 | 26.96 |
| Infinity-RoPE (Yesiltepe et al., 2026) | 84.91 | 79.28 | 96.66 | 96.75 | 99.09 | 98.61 | 64.2 | 64.11 | 70.40 | 26.40 |
| Rolling Forcing (Liu et al., 2026) | 84.26 | 69.80 | 94.80 | 95.71 | 98.93 | 98.63 | 62.2 | 62.85 | **72.34** | 24.57 |
| LongLive (Yang et al., 2026) | 84.13 | 81.11 | **97.02** | **96.95** | **99.34** | **98.80** | 46.1 | **66.97** | 69.43 | 26.62 |
| Self-Forcing++ (Cui et al., 2025) | 84.57 | 79.79 | 95.64 | 96.53 | 99.14 | 98.46 | 64.4 | 64.60 | 69.46 | 26.74 |
| Reward (Lu et al., 2026) | 85.21 | **81.39** | 95.46 | 96.65 | 98.96 | 98.33 | 72.5 | 65.71 | 69.38 | 27.04 |
| BAgger (Po et al., 2026) | 83.23 | 75.98 | 92.37 | 94.87 | 98.60 | 98.57 | **78.9** | 59.35 | 65.88 | 25.38 |
| **ARROW** | 84.89 | 80.38 | 95.18 | 96.09 | 98.99 | 98.27 | 70.0 | 65.75 | 69.81 | **27.05** |
| *MAGI-1-4.5B, distilled (Sand.ai et al., 2025)* | | | | | | | | | | |
| Native | 81.68 | 71.31 | **95.60** | 96.23 | 99.00 | **99.20** | 43.3 | 62.75 | **61.31** | 24.30 |
| **ARROW** | **81.69** | **71.33** | 95.55 | **96.30** | **99.02** | 99.16 | **43.6** | **62.86** | 61.21 | **24.34** |
| *HunyuanVideo1.5-8B / minWM (Wu et al., 2025; Zhao et al., 2026)* | | | | | | | | | | |
| Native | 85.12 | 81.39 | 96.63 | **97.13** | 99.17 | **98.85** | 53.1 | 67.41 | **72.50** | 26.96 |
| **ARROW** | **85.13** | **81.44** | **96.67** | 97.11 | **99.18** | 98.82 | **53.3** | **67.51** | 72.40 | **27.01** |

Table 8: **VideoPhy-2 on Wan.** Three-seed means: semantic adherence (SA), physical commonsense (PC), and joint success (both ratings $\geq 4$). Five-second requests; bold marks optima. Methods follow Table 1.

| Method | SA | PC | Joint (%) |
|---|---|---|---|
| Native | 3.498 | 2.843 | 20.42 |
| Deep Forcing (Yi et al., 2026) | 3.521 | 2.846 | 19.91 |
| Infinity-RoPE (Yesiltepe et al., 2026) | 3.470 | 2.908 | 22.11 |
| Rolling Forcing (Liu et al., 2026) | 3.318 | 2.769 | 17.03 |
| LongLive (Yang et al., 2026) | 3.566 | **2.913** | 22.22 |
| Self-Forcing++ (Cui et al., 2025) | 3.544 | 2.874 | 21.38 |
| Reward (Lu et al., 2026) | **3.585** | 2.865 | **22.45** |
| BAgger (Po et al., 2026) | 3.412 | 2.781 | 18.61 |
| **ARROW** | 3.506 | 2.850 | 20.81 |

Table 9: **VMBench on Wan.** Three-seed, five-second means on the 100-point scale. CAS/MSS/OIS/PAS/TCS measure commonsense, smoothness, integrity, amplitude and coherence. Bold marks optima; methods follow Table 1.

| Method | CAS | MSS | OIS | PAS | TCS |
|---|---|---|---|---|---|
| Native | 48.50 | 60.86 | 72.45 | 26.00 | 90.19 |
| Deep Forcing (Yi et al., 2026) | 48.94 | 60.35 | 72.80 | 27.75 | 90.14 |
| Infinity-RoPE (Yesiltepe et al., 2026) | 47.94 | 62.50 | 74.20 | 24.80 | 91.59 |
| Rolling Forcing (Liu et al., 2026) | 46.14 | 62.20 | 73.44 | 23.26 | 91.04 |
| LongLive (Yang et al., 2026) | 49.07 | **62.85** | **75.39** | 24.22 | **92.02** |
| Self-Forcing++ (Cui et al., 2025) | 49.02 | 61.53 | 73.50 | 26.53 | 91.11 |
| Reward (Lu et al., 2026) | **50.52** | 60.29 | 71.74 | 30.23 | 89.85 |
| BAgger (Po et al., 2026) | 46.21 | 59.43 | 70.04 | **32.87** | 87.87 |
| **ARROW** | 48.60 | 60.76 | 72.54 | 25.90 | 90.29 |

On VideoPhy-2 (Table 8), the semantic and physical ratings of the matched pair change by less than 0.01, and joint success rises by 0.39 percentage points (95% CI 0.11 to 0.73). Reward Forcing has the highest joint success and LongLive the highest physical rating. Joint success requires a correct and physically plausible action within five seconds, which depends mainly on the trained model. Across the external systems, joint success ranges from 17.03 to 22.45, a spread more than ten times the matched ARROW effect, so training choices dominate this five-second metric.

Table 9 reports the five VMBench metrics. LongLive leads MSS, OIS and TCS, Reward Forcing leads CAS and BAgger leads PAS. Within the matched pair, MSS and PAS each fall by 0.10 points (95% CI $-0.11$ to $-0.09$ and $-0.12$ to $-0.09$), and the other three metrics rise slightly. PAS measures perceptible motion amplitude within five seconds; long-video Dynamic Degree in Table 1, which rises with ARROW, measures whether motion continues over a minute. BAgger and Reward Forcing, which have the highest PAS, have the two lowest OIS and TCS of all nine systems: larger motion amplitude comes with lower object integrity and temporal coherence in this suite.

Table 10 reports all seven T2V-CompBench categories. Different systems lead different categories: LongLive leads consistent-attribute binding, spatial relations and interactions, BAgger motion binding, Reward Forcing dynamic-attribute binding, Self-Forcing++ action binding and ARROW generative numeracy. The matched Wan pair differs by less than one point in each category. Motion binding decreases by 0.23 points, dynamic-attribute binding by 0.16 (95% CI $-0.33$ to $-0.02$) and object interactions by 0.21 ($-0.30$ to $-0.12$), while the other four categories rise slightly. These categories

Table 10: **T2V-CompBench with the complete Wan comparison roster.** All seven official task scores are multiplied by 100; higher is better. Five-second requests and evaluator settings are shared, while external methods retain their system settings. Cons./Dyn. denote consistent-/dynamic-attribute binding; Interact.: object interactions; Count: generative numeracy. Bold marks column optima; blue identifies ARROW. Method abbreviations follow Table 1.

| Method | Attributes ↑ | | Relations and actions ↑ | | | | Count ↑ |
|---|---|---|---|---|---|---|---|
| | Cons. | Dyn. | Spatial | Motion | Action | Interact. | |
| *Wan2.1-1.3B / Self Forcing (Huang et al., 2025; Wan Team et al., 2025)* | | | | | | | |
| Native | 59.57 | 2.25 | 50.49 | 28.15 | 52.47 | 59.52 | 35.15 |
| Deep Forcing (Yi et al., 2026) | 59.65 | 3.14 | 50.73 | 29.74 | 51.68 | 58.80 | 34.27 |
| Infinity-RoPE (Yesiltepe et al., 2026) | 60.37 | 2.80 | 50.91 | 26.21 | 51.46 | 58.68 | 34.33 |
| Rolling Forcing (Liu et al., 2026) | 56.14 | 2.03 | 48.85 | 25.77 | 46.34 | 54.48 | 32.47 |
| LongLive (Yang et al., 2026) | **61.29** | 2.56 | **51.94** | 26.41 | 53.09 | **60.44** | 35.20 |
| Self-Forcing++ (Cui et al., 2025) | 59.41 | 2.94 | 50.31 | 29.15 | **53.31** | 59.66 | 33.90 |
| Reward (Lu et al., 2026) | 59.53 | **3.45** | 51.32 | 30.83 | 53.23 | 59.51 | 34.17 |
| BAgger (Po et al., 2026) | 56.44 | 3.25 | 47.77 | **33.55** | 50.41 | 56.09 | 32.92 |
| **ARROW** | 59.65 | 2.09 | 50.55 | 27.92 | 52.69 | 59.31 | **35.38** |

require a specified event within five seconds, a movement, an attribute change or a physical interaction, which depends on following the prompt's action more than on continuity across blocks.

Across the three five-second suites, the matched Wan pair differs by less than one point on every metric, while external systems trained with their own objectives lead individual criteria. Composition, physics and motion within five seconds are therefore set mainly by the trained model, and the writing pass changes them little; its effect grows with the number of writes (Section 5.2).

Temporal metrics such as warping error combine motion and appearance change. Figure 16 separates them on one video. Optical flow measures the displacement between two frames, and the residual after warping measures what the displacement does not explain. Flow is large only on the falling water. Residuals appear along the waterfall and also along the horizon, the cliff and the ridges of the terrain, where flow is small; there they reflect texture and edge effects of the warping. A warping-based temporal score therefore mixes the motion of the water with edge effects on static terrain, and reading flow and residual together separates the two.

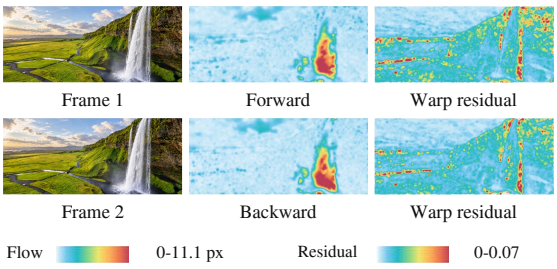



Figure 16: Motion and residual change in a waterfall video. Left: two frames. Middle: forward and backward flow magnitude (0–11.1 pixels). Right: color residual after flow warping (0–0.07).

Figure 17 varies the acceptance threshold of six criteria and reports the change in pass rate relative to native Wan. ARROW raises the temporal-consistency pass rate by up to about four points for thresholds between 0.87 and 0.93, and its curves for the other five criteria stay within about two points of zero. Reward Forcing shows the opposite profile: it gains up to about 14 points on motion binding and 10 on camera motion, and loses up to about 8 points on temporal consistency at the same thresholds where ARROW gains. Training with a motion reward thus trades consistency for motion, whereas changing the writing pass improves consistency and leaves prompt-specific actions to the trained model.

### D.5 Human Preference

Table 11 reports seven criteria from one Movie Gen voting protocol (Polyak et al., 2024). The comparison uses 1,003 prompts, three seeds and randomized left–right placement. Each pair of videos is rated by panels drawn from a pool of 24 raters: three raters judge text alignment, visual quality, frame consistency, motion completeness and motion naturalness, and six judge realness and aesthetics. A strict majority determines a win or a loss, and other outcomes are ties. Net preference averages wins minus losses over paired videos.

Against native Wan, ARROW is preferred on every criterion, most strongly on frame consistency (43.34) and visual quality (38.78). Against the seven external systems, its net preference is positive on every criterion with one exception, motion completeness: Reward Forcing and BAgger are preferred

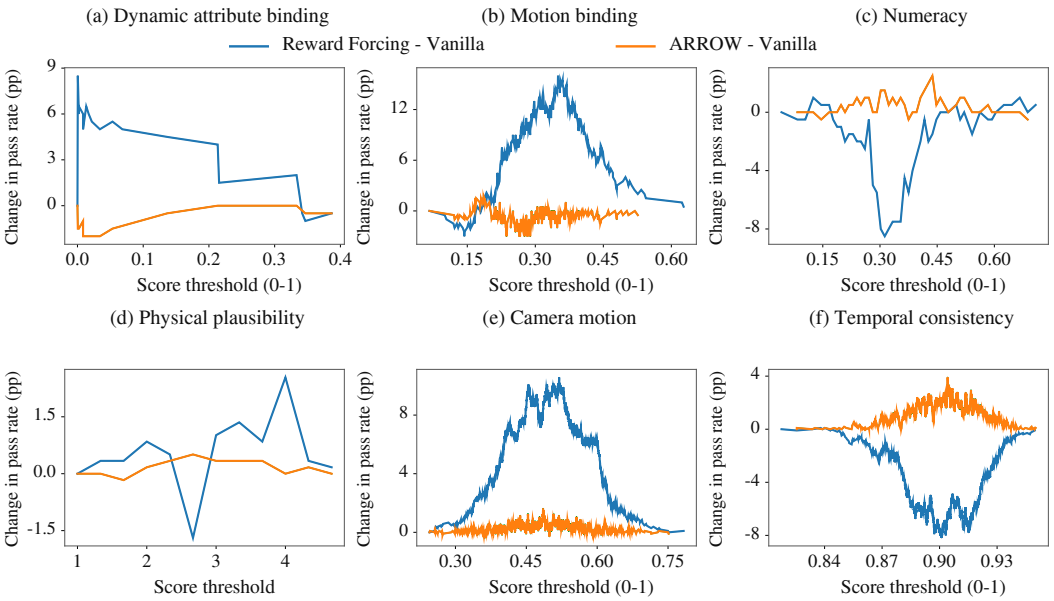



Figure 17: Change in pass rate relative to native Wan (Vanilla) as the acceptance threshold increases, for ARROW and Reward Forcing: (a) dynamic-attribute binding, (b) motion binding, (c) numeracy, (d) physical plausibility, (e) camera motion and (f) temporal consistency. Thresholds follow each criterion's native scale; zero denotes no change.

Table 11: **ARROW preference over each Wan opponent.** Mean majority-win minus majority-loss points on Movie Gen; positive favors ARROW. Visual/Cons.: visual quality/frame consistency; Complete/Natural: motion completeness/naturalness. Bold marks the largest net contrast. Appendix D.5 gives rater-sensitive intervals; methods follow Table 1.

| Opponent | Text | Appearance | | | | Motion | |
|---|---|---|---|---|---|---|---|
| | | Visual | Cons. | Realness | Aesthetics | Complete | Natural |
| *Wan2.1-1.3B (Huang et al., 2025; Wan Team et al., 2025); Movie Gen prompts (Polyak et al., 2024)* | | | | | | | |
| Native | **32.20** | **38.78** | **43.34** | 30.18 | 23.26 | **26.69** | **30.31** |
| Deep Forcing (Yi et al., 2026) | 22.03 | 24.19 | 24.66 | 17.88 | 15.82 | 14.79 | 19.91 |
| Infinity-RoPE (Yesiltepe et al., 2026) | 21.27 | 17.31 | 16.75 | 20.77 | 15.92 | 21.17 | 16.55 |
| Rolling Forcing (Liu et al., 2026) | 27.12 | 19.44 | 21.54 | 19.44 | 11.76 | 26.29 | 21.14 |
| LongLive (Yang et al., 2026) | 19.87 | 15.75 | 11.93 | 12.30 | 6.71 | 20.34 | 14.89 |
| Self-Forcing++ (Cui et al., 2025) | 16.88 | 15.19 | 12.10 | 13.06 | 11.10 | 12.30 | 13.16 |
| Reward (Lu et al., 2026) | 14.19 | 15.09 | 20.80 | 17.65 | 15.42 | -0.96 | 15.65 |
| BAgger (Po et al., 2026) | 24.69 | 31.54 | 31.37 | **31.11** | **26.39** | -1.69 | 18.38 |

by 0.96 and 1.69 points, with intervals for ARROW minus opponent of $-10.83$ to 9.94 and $-10.83$ to 9.14. These two systems also have the highest Dynamic Degree in Table 1. Raters therefore distinguish a consistent video from a complete action, and ARROW improves the first.

The two-way bootstrap draws multinomial weights for prompts and raters independently, recomputes each weighted panel majority and averages over 10,000 resamples; an empty panel counts as a tie. This keeps the dependence induced by reused raters. With three-vote panels, the majority is a nonlinear function of the weights, so the percentile interval can shift away from the point estimate: against native Wan, the visual-quality estimate is 38.78 and its 95% interval is 14.56 to 37.19. Visual-quality intervals include zero against Infinity-RoPE ($-0.03$ to 22.37), LongLive ($-0.96$ to 22.00), Self-Forcing++ ($-2.03$ to 21.37) and Reward Forcing ($-1.23$ to 22.33).

Figure 18 illustrates the two criteria. Reading down a column follows the action from start to end; reading across compares the subject and background between systems. The x–t strips along a fixed scanline show how steadily the subject and background colors persist over time. The distinction matches Table 11: consistency of subject and background and completion of the action are judged separately.

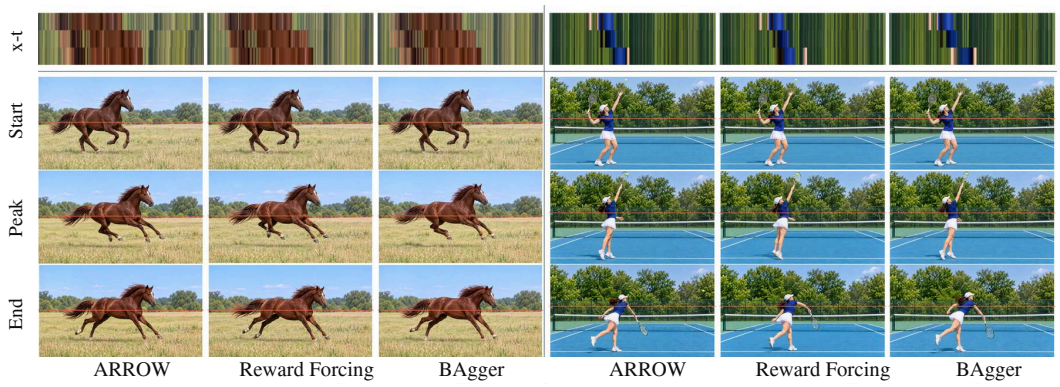



Figure 18: Appearance and action progression for a running horse and a tennis serve, from ARROW, Reward Forcing and BAgger at the start, peak and end of the action. The x–t strips above are taken along the fixed red scanline.

## E Additional Visual Comparisons

The examples below complement the one-minute comparison of Figure 5 with four further ARROW videos and a repeated action across five systems. Time runs from left to right within a row, and conditions are compared at matching columns.

### E.1 ARROW over One Minute

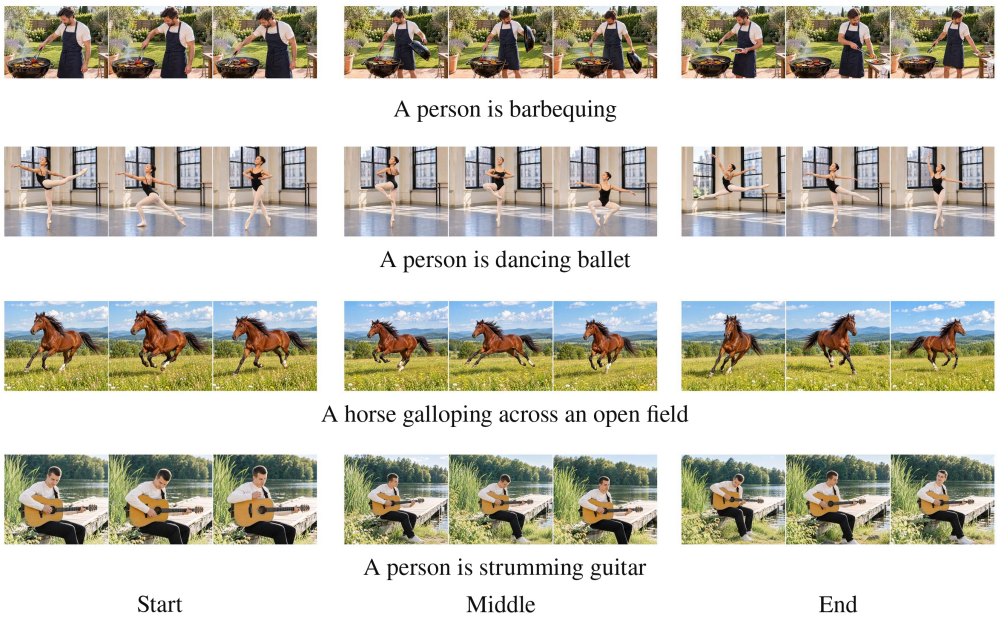



Figure 19: ARROW on Wan over one minute. Each row shows three frames each from the start, middle and end of one video: barbecuing, ballet, a galloping horse and guitar playing. Compare the person's clothing and the scene layout across the three groups.

Figure 19 shows four one-minute ARROW videos on Wan. Each video keeps its subject's clothing and the layout of its scene, and the action continues: the barbecue is tended, the dancer moves through several poses, the horse keeps galloping and the guitarist keeps playing. These are the properties measured by the consistency scores and Dynamic Degree in Table 1.

### E.2 Appearance and Action over One Minute




Figure 20: Basketball shooting over one minute on Wan from the native model, Deep Forcing, Rolling Forcing, Reward Forcing and ARROW, at 0, 8, . . . , 48 and 60 s. Each row is one video.

Figure 20 shows one prompt with a repeated action. In every system the player dribbles, bends for the ball and shoots several times, so the progression of the action is comparable across rows. The appearance of the player, the court markings and the buildings or trees behind the court can then be followed at the same eight times. This separates the two properties that Appendix D.5 rates separately, frame consistency and motion completeness.