HEADCAST: CASTING ATTENTION HEADS FOR EFFICIENT AUTOREGRESSIVE VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive (AR) video diffusion models have become a promising paradigm for long and streaming video synthesis, but the continuously growing Key-Value (KV) cache makes attention the dominant inference cost. This cost is governed by how many tokens sit in the cache, so it becomes the bottleneck precisely in the regime the field is moving toward: high resolution, where each frame contributes many tokens, and long horizons, where frames accumulate. Existing remedies either evict the cache with coarse heuristics that cause inter-frame flickering, or require model re-training. We show that the attention heads of a pre-trained AR video DiT possess a *stable structural property*—each head's spatio-temporal receptive field is reproducible from a small, fixed sub-context, and which sub-context suffices can be identified from a single forward pass. We formalize this as four archetypes—**Sink**, **Dummy**, **Spatial**, and **Global**—and build **HeadCast**, a training-free, plug-and-play framework that classifies every head once at the maximum-noise step and restructures the monolithic KV cache into head-specific pathways. Crucially, it retains the Global heads that preserve the long-range temporal consistency aggressive eviction destroys. Because the Spatial pathway operates on a fixed-size grid, its savings grow monotonically with the token count: across state-of-the-art AR models, end-to-end speedup rises from 1.02–$1.08\times$ at 480P to **1.62**$\times$ at 720P and **1.95**$\times$ at 1080P, entirely without training, while keeping VBench quality within 0.2 points of full attention and largely flicker-free. A single threshold set transfers unchanged across four backbones, three resolutions and both clip lengths (5 s and 30 s), indicating that the taxonomy reflects model-independent structure rather than per-checkpoint calibration.

1 INTRODUCTION

Diffusion Transformers (Peebles & Xie, 2023) have advanced high-fidelity video generation (Ho et al., 2022; Blattmann et al., 2023; Yang et al., 2025b; Kong et al., 2024; Wan Team, 2025). Rather than generating all frames at once, autoregressive (AR) models synthesize video block-by-block, supporting long-horizon and streaming generation by reusing historical states through Key-Value (KV) caching. But the cache grows without bound, and attention over it incurs an $\mathcal{O}(L^2)$ cost that comes to dominate inference. A sliding window bounds this growth, yet at high resolution the in-window sequence is still long, so inference remains slow; and shrinking the cache aggressively is no remedy either—Dummy Forcing (Guo et al., 2026) evicts long-range context that some heads depend on, producing flickering and structural drift, while training-based sparse attention requires costly re-training.

It matters where this bottleneck bites. Attention's share of inference is set by the number of cached tokens, which grows with resolution (quadratically in the spatial grid) and with rollout length. At 480P over a few seconds the cache is small and attention is not yet dominant, so *no* cache-compression method—ours included—has much headroom; the cache becomes the binding constraint only in the regime the field is moving into, high-resolution and minute-scale generation (Yang et al., 2026; Cui et al., 2025). We target it explicitly and report the full resolution sweep rather than a single headline number, so the trend—and its absence at 480P—stays visible.

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.

This raises a question: must every head attend to the full historical context, or do heads differ in what they need? Visualizing pre-trained AR video models, we find stable, heterogeneous structural preferences falling into four archetypes: **Sink** heads anchor on the initial frame, **Dummy** heads attend only to the most recent block, **Spatial** heads attend to a local spatial neighborhood across history, and **Global** heads need the full context. These assignments persist throughout generation, so one early classification suffices.

We therefore propose **HeadCast**, a training-free, plug-and-play framework that *casts* each head to a dedicated computation path. After a brief full-context warm-up it classifies every head once at the maximum-noise step ($t = 1000$), where attention reflects structural rather than content-specific preferences, and routes it accordingly: Sink and Dummy heads keep a single block, Spatial heads attend within a fixed grid, and Global heads retain the full sliding window, with the monolithic cache split into compact head-specific buffers. Because the Spatial pathway is confined to a fixed-size grid, the speedup grows with the cache, reaching $1.62\times$ at 720P and $1.95\times$ at 1080P without training. Across Self-Forcing (Huang et al., 2025), LongLive (Yang et al., 2026), Causal Forcing (Zhu et al., 2026), and Reward Forcing (Lu et al., 2026), it preserves VBench quality and frame-level fidelity while removing the flickering eviction induces.

In summary, our main contributions are as follows:

- We show that head-level locality in pre-trained AR video DiTs is a *stable, one-shot-identifiable structural property*: four archetypes—Sink, Dummy, Spatial, Global—suffice to describe it, a single forward pass at maximum noise recovers the assignment, and one threshold set transfers unchanged across four backbones, three resolutions and both clip lengths (5 s and 30 s) with no retuning.
- We turn this into HeadCast, a training-free framework that splits the monolithic KV cache into head-specific buffers and routes each head to a matching attention operator, realized as dense sub-matrices batched through FlashAttention so that nominal sparsity becomes real wall-clock speedup; Global heads are explicitly retained, which is what preserves temporal consistency.
- We show the payoff grows monotonically with the cached-token count—1.02–$1.08\times$ at 480P, $1.62\times$ at 720P, $1.95\times$ at 1080P, end-to-end—while preserving fidelity and temporal coherence, and verify by an iso-throughput frontier comparison that this fidelity comes from the taxonomy rather than from a milder compression rate.

2 Related Work

2.1 Autoregressive Video Diffusion

Diffusion models (Ho et al., 2020; Song et al., 2021; Rombach et al., 2022) dominate high-fidelity visual synthesis, and a growing line of work (Chen et al., 2024; Yin et al., 2025; Huang et al., 2025; Liu et al., 2026; Yang et al., 2026; Sand AI, 2025) combines them with autoregressive (AR) prediction for long-horizon, streaming generation, cutting cost through causal modeling and KV caching: CausVid (Yin et al., 2025) makes a bidirectional DiT causal with KV caching, Self-Forcing (Huang et al., 2025) removes the train-inference mismatch by conditioning on its own generated frames, Rolling Forcing (Liu et al., 2026) widens the diffusion window against error accumulation, and LongLive (Yang et al., 2026) adds KV re-caching for continuity across scene transitions. Such generators are often further distilled into few-step samplers (Song et al., 2023; Yin et al., 2024; Salimans & Ho, 2022), which cuts the *number* of denoising steps; HeadCast is orthogonal and training-free, reducing the per-step attention cost instead.

2.2 KV Cache Compression

Attention's (Vaswani et al., 2017) linearly growing memory footprint has motivated extensive KV cache compression, primarily for LLMs: token-level methods preserve initial-token "sinks" (Xiao et al., 2024) or keep pivotal tokens by attention score (Zhang et al., 2023; Li et al., 2024), head-specific methods (Xiao et al., 2025; Ge et al., 2024) assign different cache *budgets* per head, and complementary work attacks the cost with IO-aware kernels (Dao et al., 2022) or low-bit quantization (Liu et al., 2024). All target the 1D structure of text and ignore video's spatio-temporal redundancy, and a scalar per-head budget is weaker than identifying *which geometry* a head needs.

HeadCast extends head-level restructuring to AR video diffusion with dimension-aware, per-type policies.

2.3 Efficient Video Generation

Early video models adopted bidirectional Diffusion Transformers (DiT) (Peebles & Xie, 2023), accelerated mainly via fixed-length sparse attention (Zhang et al., 2025b;a). The Sparse VideoGen family (Xi et al., 2025; Yang et al., 2025a) identifies spatial and temporal heads (clustered token blocks in SVG2), and Sparse-vDiT (Chen et al., 2026) profiles per-layer and per-head patterns to pick a strategy for each—but all are defined under *bidirectional* 3D full attention, where every head sees a fixed, complete spatio-temporal volume. They do not transfer to the causal, dynamically growing context of AR models, where a head's dependencies must be characterized against an ever-lengthening history with no future frames.

Closer to our setting, Light Forcing (Lv et al., 2026) allocates chunk-level sparsity by coarse-to-fine top-k but applies one scheme to all heads, and Sparse Forcing (Xu et al., 2026) learns native sparse attention—both require training. A training-free line compresses the cache directly: the closest, Dummy Forcing (Guo et al., 2026), forces heads with only mild temporal sparsity into an aggressive "dummy" mode keeping just the current frame, over-discarding context they depend on and inducing flickering; Deep Forcing (Yi et al., 2026) is training-free but targets long-range stability rather than acceleration; concurrent Head Forcing (Tian et al., 2026) exploits head heterogeneity for *long-horizon extrapolation* via hierarchical memory and head-wise RoPE re-encoding; and concurrent Forcing-KV (Ji et al., 2026) has no public implementation. HeadCast instead adds a *Spatial* type for spatial locality and explicitly retains a *Global* type for the rest, attaining far higher fidelity than Dummy Forcing at modest speed cost.

3 Motivation

3.1 Preliminary: Autoregressive Video Diffusion and KV Cache

Autoregressive video diffusion models generate video *block-by-block*: a video of T frames is grouped into B blocks $X = \{X_1, \ldots, X_B\}$ of b frames each, and the model learns the joint distribution under a *block*-causal factorization matching how generation proceeds:

$$p(X_1, \ldots, X_B) = \prod_{i=1}^{B} p(X_i \mid X_{<i}) \tag{1}$$

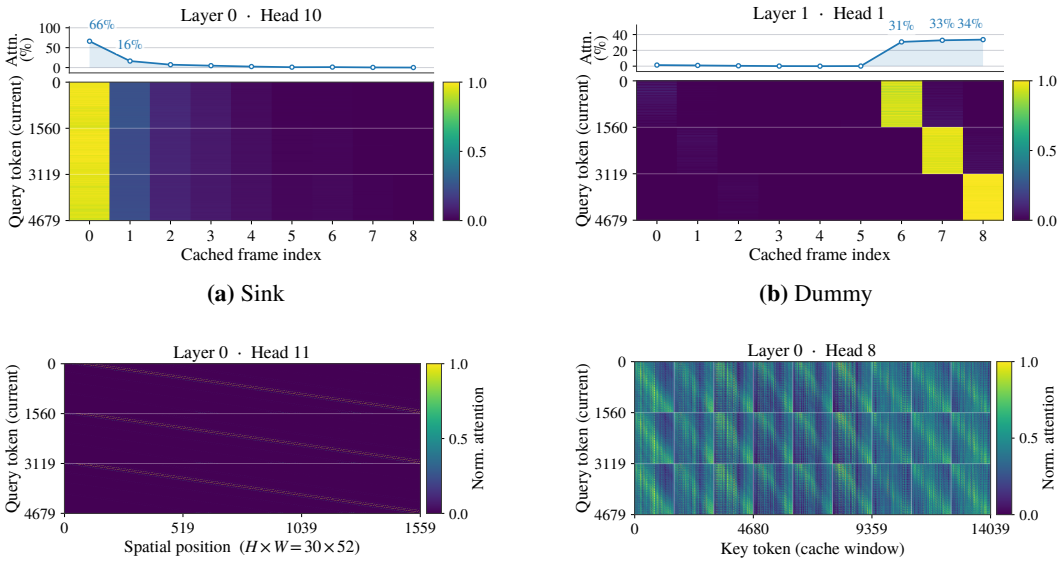
Each conditional is a denoising diffusion model (Peebles & Xie, 2023) that denoises all b frames of X_i jointly given the previous blocks, so frames within a block attend bidirectionally while the block-causal mask forbids attention to future blocks—which is what makes historical keys and values (KV) reusable across AR steps. Throughout, *full attention* denotes dense attention over the baseline's bounded sliding window (not the entire history), and *full history* the whole past.

A sliding window of recent frames (Huang et al., 2025) curbs the resulting $\mathcal{O}(L^2)$ cost but exposes a tension: at high resolution the in-window sequence is still long enough to keep inference slow, yet shrinking the window discards long-range context longer videos need for coherence. That retaining the first frame as a fixed sink stabilizes quality already hints not all cached content matters equally—but Dummy Forcing (Guo et al., 2026) evicts too coarsely, dropping context heads depend on. Efficiency without quality loss thus demands a classification aligned with each head's true dependencies.

3.2 Heterogeneous Attention Patterns in Video DiT

In current AR video diffusion models, every attention head attends over the entire accumulated history. We hypothesize that this is unnecessary—heads have intrinsically diverse dependencies on that history—and visualize the attention maps of an AR baseline (e.g., Self-Forcing) during inference to expose their heterogeneous patterns.

Observation 1: heads are sparse in the temporal scope. Following Dummy Forcing (Guo et al., 2026), we aggregate each query's attention over all key tokens in the same frame to quantify per-



(a) Sink **(b)** Dummy

(c) Spatial **(d)** Global

Figure 1: The four attention-head archetypes in a pre-trained AR video model. **(a)** Sink and **(b)** Dummy heads (per-frame view) concentrate on the first and the most recent block, respectively; **(c)** a Spatial head (per-position view) attends to each query's local neighborhood; **(d)** a Global head (raw map) spreads across the whole history.

frame importance (Figure 1(a)–(b)). Many heads are markedly sparse along time, in two patterns: a **Sink Pattern**, in which the attention mass of all query tokens congregates on the *first* block of the KV cache; and a **Dummy Pattern**, in which it concentrates almost exclusively on the *current* denoising block.

Observation 2: heads are sparse in the spatial scope. For heads without extreme temporal sparsity, we project the cache onto the 2D spatial grid and aggregate attention per position, revealing each query's spatial receptive field. This splits the remainder into a **Spatial Pattern**, where a query's attention congregates on keys inside its own localized spatial neighborhood across all historical frames (Figure 1(c)); and a **Global Pattern**, where attention spreads over the entire KV cache along both axes (Figure 1(d)), so these heads genuinely need the full history.

Observation 3: the pattern of a head is stable over inference. Most heads keep their archetype across AR steps, denoising timesteps, and prompts: over **90**% receive the *identical* archetype across AR steps (frame 12 vs. 18), with similar stability along the other two axes (per-axis rates in Appendix B). A single early classification therefore suffices, at no recurring cost.

4 HeadCast

4.1 Overview

HeadCast runs in four phases (Figure 2). A brief *Warm-up* accumulates a stable full context; once the cache exceeds the window size W, a one-time *Online Classification* sorts every head into Sink, Dummy, Spatial, or Global from the cosine similarity between its full- and restricted-context outputs. *Heterogeneous Cache Management* then splits the monolithic cache into type-specific buffers along the head dimension, and *Head-Specific Attention* routes each head through its own path, cutting both sequence length and attention FLOPs.

4.2 Online Classification

Classification fires once the accumulated KV cache first exceeds the window size W, at which point the *Warm-up* has produced the full historical context that serves as the reference.

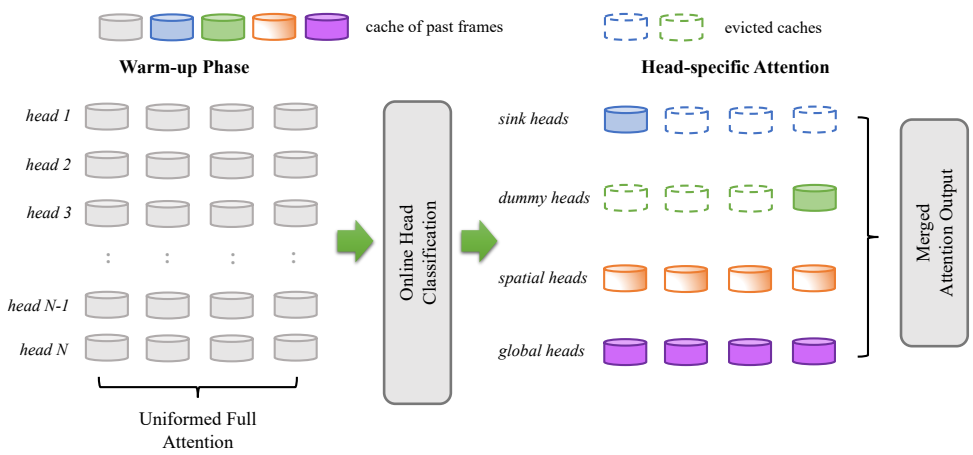


Figure 2: The overall four-phase execution pipeline of the proposed HeadCast framework, spanning from the initial full-context Warm-up and Online Classification to the downstream Heterogeneous Cache Management and Head-Specific Attention.

4.2.1 Classification Metrics and Archetypes

Given the queries Q and the full historical key-value pairs $(K_{\text{full}}, V_{\text{full}})$, we compute the unconstrained reference output of each head h:

$$O_{\text{ref}}^{(h)} = \text{Attention}\left(Q^{(h)}, K_{\text{full}}^{(h)}, V_{\text{full}}^{(h)}\right) \tag{2}$$

We then define three restricted attention operations over designated subsets of the history:

- **Sink Proxy:** Attention restricted to the first temporal block (the semantic anchor), yielding $O_{\text{sink}}^{(h)}$.
- **Dummy Proxy:** Attention restricted to the current local block of most recent frames, yielding $O_{\text{dummy}}^{(h)}$.
- **Spatial Proxy:** Attention over all historical blocks but confined to a local $(2r+1) \times (2r+1)$ grid centered on each query token, yielding $O_{\text{spatial}}^{(h)}$.

For each head h and proxy m, let $\mathbf{c}_m^{(h)} \in \mathbb{R}^{L_q}$ collect the *token-wise* cosine similarities between the restricted and reference outputs, $\left[\mathbf{c}_m^{(h)}\right]_t = \cos(O_m^{(h)}[t], O_{\text{ref}}^{(h)}[t])$ for $t = 1, \ldots, L_q$ and $m \in \{\text{sink}, \text{dummy}, \text{spatial}\}$, and write $\bar{c}_m^{(h)} = \frac{1}{L_q}\sum_t \left[\mathbf{c}_m^{(h)}\right]_t$ for its mean—how well the restricted context reproduces the head's output on average. Because this mean can stay high even when a head fails at a few boundary tokens (which leaves residual ghosting if such heads are routed to the Spatial path), for the spatial case we score the *worst* tokens, on the full vector rather than its mean:

$$\text{score}^{(h)} = \mathcal{P}_5\left(\mathbf{c}_{\text{spatial}}^{(h)}\right) - \gamma \cdot \text{MSE}\left(O_{\text{spatial}}^{(h)}, O_{\text{ref}}^{(h)}\right) \tag{3}$$

where $\mathcal{P}_5(\cdot)$ is the 5th percentile of its argument, so the score is governed by the worst-approximated tokens, and γ weights the MSE penalty (taken over all tokens and channels and normalized by their count). We use $\gamma = 5$ and proxy radius $r = 2$ (a 5×5 neighborhood) throughout.

4.2.2 Mutually Exclusive Decision Rules

Each head is assigned a single mode by a mutually exclusive hierarchy with thresholds $(\theta_s, \theta_d, \theta_{\text{sp}}, \theta_{\text{sc}})$:

$$\mathcal{C}(h) = \begin{cases} \text{Sink} & \text{if } \bar{c}_{\text{sink}}^{(h)} \geq \theta_s \\ \text{Dummy} & \text{elif } \bar{c}_{\text{dummy}}^{(h)} \geq \theta_d \\ \text{Spatial} & \text{elif } \bar{c}_{\text{spatial}}^{(h)} \geq \theta_{\text{sp}} \text{ and } \text{score}^{(h)} \geq \theta_{\text{sc}} \\ \text{Global} & \text{otherwise} \end{cases} \tag{4}$$

This classification runs only once, on a single autoregressive block, so it adds a one-time cost with no steady-state overhead—and it is already contained in every speedup we report (Appendix F quantifies its share: 5–8% of the total budget on 5-second clips, under 1.5% on 30-second LongLive).

4.3 Heterogeneous Cache Management

With the H heads partitioned into N_{sink}, N_{dummy}, N_{spatial} and N_{global}, the monolithic KV cache is decoupled along the head dimension into three independent caches. The **Sink-Dummy Cache** merges both temporal types to halt growth—Sink heads freeze the initial block, Dummy heads keep only the most recent block by rolling overwrite. The **Global Cache** is a standard sliding-window layout, retaining the initial sink tokens while evicting intermediate frames outside the window. The **Spatial Cache** shares that temporal layout, but the localized spatial mask of the next phase sharply reduces its effective compute. This removes memory redundancy and yields contiguous layouts for parallel computation.

4.4 Head-Specific Attention

At each AR step a lightweight kernel slices the queries along the head dimension and routes each group to its own operator. The **Sink and Dummy Paths** both attend to exactly one historical block—the initial anchor for Sink, the latest for Dummy—so their KV sequences share a shape, letting us fuse the two types into one kernel; the **Global Path** is standard sliding-window attention over the Global Cache, bounding the temporal context to W. The **Spatial Path** accesses the full window in time but under a block-to-block spatial constraint: the frame is partitioned into non-overlapping grids and a query attends only to KV tokens in its own grid across all frames in the window (illustrated in Appendix C), which decomposes the global attention matrix into many small independent submatrices that we stack along the batch dimension and run through the standard FlashAttention kernel. This deployed grid intentionally differs from the sliding $(2r{+}1) \times (2r{+}1)$ neighborhood of the Spatial *proxy*: a per-query sliding window cannot use FlashAttention and would need a slow custom FlexAttention kernel, whereas the non-overlapping grid is a hardware-efficient approximation of the same locality. The pathway outputs are finally concatenated back along the head dimension.

5 Experiments

5.1 Experimental Setup

Models and Baselines. We plug HeadCast into four recent AR video diffusion models—Self-Forcing (Huang et al., 2025), LongLive (Yang et al., 2026), Causal Forcing (Zhu et al., 2026), Reward Forcing (Lu et al., 2026)—comparing against each model's full sliding-window attention and against Dummy Forcing (Guo et al., 2026), to our knowledge the only directly comparable baseline here: Light Forcing (Lv et al., 2026) and Sparse Forcing (Xu et al., 2026) require training; the Sparse VideoGen family (Xi et al., 2025; Yang et al., 2025a) and Sparse-vDiT (Chen et al., 2026) define sparsity under *bidirectional* 3D full attention, which does not transfer to the causal, growing context of AR models; Deep Forcing (Yi et al., 2026) targets stability, not acceleration; and concurrent Forcing-KV (Ji et al., 2026) has no public implementation. We therefore also situate HeadCast against its own frontier (§5.6).

Clip lengths. Of the four backbones, only LongLive is designed for—and stable over—minute-scale rollouts, adding KV re-caching for continuity across long horizons; the other three degrade severely beyond their training horizon and eventually collapse, so a 30-second comparison on them would measure backbone failure rather than our effect. We therefore use 5-second clips for Self-Forcing, Causal Forcing, and Reward Forcing, and LongLive for the 30-second setting; since resolution and rollout length enlarge the cache independently, we vary resolution on *both* Self-Forcing and LongLive (§5.3) to keep the two trends separable.

Evaluation Metrics. We report VBench (Huang et al., 2024) on 5-second clips and VBench-Long on 30-second clips (the 128 MovieGen (Polyak et al., 2024) prompts of Self-Forcing++ (Cui et al., 2025)), refined with Qwen2.5-7B-Instruct (Yang et al., 2024) following Self-Forcing, averaged over 5 seeds per prompt. Lacking ground truth, we take each model's own full-attention output as ref-

Table 1: Main results on four autoregressive video models at standard 480P resolution (clip length shown per backbone). We report VBench (Quality, Semantic, Total) and Dynamic Degree, with PSNR and LPIPS computed against each model's full-attention output (hence "—" for the baseline). **Bold marks the better of the two acceleration methods** within each backbone block (omitted on ties); the full-attention row (⋄) is the reference and is never bolded, so bold does *not* claim superiority over full attention. Dynamic Degree has no preferred direction—closeness to the baseline is what matters—so it is unbolded.

Method	FPS ↑	Speedup ↑	Quality ↑	Semantic ↑	Total ↑	Dynamic Degree	PSNR ↑	LPIPS ↓
⋄ **Self-Forcing** (5s)	19.48	1.00×	84.60	80.77	83.84	64.72	—	—
• + Dummy Forcing	20.30	1.04×	84.22	80.77	83.53	55.83	18.23	0.1919
• **+ HeadCast**	20.38	**1.05**×	**84.66**	80.77	**83.88**	64.72	**25.64**	**0.0469**
⋄ **LongLive** (30s)	16.06	1.00×	82.77	80.17	82.26	35.59	—	—
• + Dummy Forcing	18.63	**1.16**×	**82.93**	80.69	**82.48**	34.81	10.27	0.6161
• **+ HeadCast**	17.29	1.08×	82.42	**80.93**	82.12	29.84	**11.02**	**0.5779**
⋄ **Causal Forcing** (5s)	20.08	1.00×	85.42	80.95	84.53	85.0	—	—
• + Dummy Forcing	20.37	1.01×	84.58	80.86	83.83	94.6	15.96	0.2237
• **+ HeadCast**	20.46	**1.02**×	**85.37**	**81.01**	**84.50**	95.2	**22.42**	**0.0678**
⋄ **Reward Forcing** (5s)	19.81	1.00×	84.94	80.88	84.13	68.33	—	—
• + Dummy Forcing	20.17	1.02×	84.64	**80.90**	83.89	58.05	18.63	0.1507
• **+ HeadCast**	20.14	1.02×	**84.79**	80.87	**84.00**	65.56	**24.92**	**0.0449**



Figure 3: Inter-frame flickering at an AR block boundary (LongLive, 30 s, 720P). Frames 127–131 span a block edge at frame 129. Dummy Forcing's aggressive cache eviction injects a spurious structure—a tree abruptly appears on the right (red box) at frame 130—while *Full Attention* and our *HeadCast* keep the scene temporally consistent.

erence and report PSNR/LPIPS (Zhang et al., 2018) against it as a *fidelity* axis complementing VBench's *absolute* one. All timings use a single GPU.

Implementation Details. Baseline and HeadCast share the same sliding window (3 frames per block, 1 fixed sink block): 4 blocks for 5-second clips, 7 for 30-second. Classification fires once the window first fills (e.g. $S_{\text{start}} = 7$ for 30-second clips) at $t = 1000$; the cosine thresholds share one value $\theta_s{=}\theta_d{=}\theta_{\text{sp}}{=}0.95$, the spatial score threshold is $\theta_{\text{sc}}{=}0.755$, and the Spatial path uses non-overlapping 10×10 cells—one set for *every* backbone and resolution (§5.5). **All reported FPS and speedups are end-to-end wall-clock** over the complete generation, already including warm-up and the one-time classification; Appendix F reports its share.

5.2 Main Results

Table 1 compares HeadCast, Dummy Forcing, and full attention across the four backbones. On VBench Total, HeadCast stays within 0.15 points of the baseline everywhere, whereas Dummy Forcing loses up to 0.7. Dynamic Degree (a coarse motion indicator) varies by backbone: HeadCast matches the baseline on Self-Forcing, stays close on Reward Forcing, is lower on LongLive (29.84 vs. 35.59), and reads *higher* on Causal Forcing. The methods diverge far more sharply on frame-level fidelity, which VBench overlooks: HeadCast reaches 22–26 dB PSNR on the three 5-second backbones against Dummy Forcing's 16–19 dB, cutting LPIPS by 3–4×. That gap is exactly the flickering its eviction introduces—structures popping in and out at block boundaries (Figure 3)—which HeadCast avoids by retaining the Global heads; a block-boundary discontinuity metric (Appendix G) confirms it, Dummy Forcing inflating the block-edge jump by up to 36% on long clips while HeadCast matches the baseline.

Method	Res.	FPS↑	Spd.↑	Total↑	PSNR↑	LPIPS↓
Self-Forcing (5s)	720P	7.04	1.00×	83.78	—	—
+ Dummy Forcing	720P	9.60	1.36×	**83.91**	18.81	0.2021
+ HeadCast	720P	9.21	1.31×	83.83	**25.87**	**0.0568**
LongLive (30s)	720P	4.83	1.00×	79.07	—	—
+ Dummy Forcing	720P	8.40	1.74×	78.85	12.83	0.4554
+ HeadCast	720P	7.82	1.62×	**79.05**	**15.22**	**0.3357**
Self-Forcing (5s)	1080P	1.64	1.00×	82.80	—	—
+ Dummy Forcing	1080P	2.68	1.63×	82.66	20.24	0.2059
+ HeadCast	1080P	2.40	1.46×	**82.90**	**27.17**	**0.0693**
LongLive (30s)	1080P	1.07	1.00×	76.84	—	—
+ Dummy Forcing	1080P	2.31	2.16×	77.18	12.29	0.5394
+ HeadCast	1080P	2.09	1.95×	**77.26**	**14.35**	**0.4198**

Table 2: Scalability to 720P and 1080P on Self-Forcing (5-second clips) and LongLive (30-second clips). PSNR and LPIPS are computed against each model's full-attention output (hence "—" for the baseline row, which is its own reference). As in Table 1, **bold marks the better of the two acceleration methods** within each block; the baseline is the reference and is never bolded, so bold does *not* claim superiority over full attention.

Comparison	Ours	Opp.
vs. Full Attn.	**39.6%**	60.4%
vs. Dummy F.	**90.9%**	9.1%

Table 3: User study on LongLive (30 s, 720P). Percentage of 2AFC comparisons preferring each method (each row sums to 100%).

Configuration	Temp.	Spat.	Spd.↑	PSNR↑	VBench↑
All Global (Baseline)	×	×	1.00×	—	83.78
+ Spatial only	×	✓	1.19×	25.04	83.76
+ Sink+Dummy only	✓	×	1.04×	27.99	83.82
Full HeadCast	✓	✓	**1.31**×	25.87	**83.83**

Table 4: Ablation on head archetypes (Self-Forcing, 5 s, 720P). Temp. groups Sink and Dummy heads; Spat. denotes Spatial heads.

These fidelity gains cost essentially nothing in speed. At 480P the cache is small, so attention is not yet dominant and both methods gain little (HeadCast 1.02–1.08×, Dummy Forcing 1.01–1.16×); only on LongLive (30 s) does Dummy gain a meaningful FPS lead, and still at 0.75 dB lower PSNR. The real gains come at high resolution (§5.3), where the cache grows and HeadCast's savings scale with it.

5.3 Scalability to High Resolution

HeadCast's savings grow with resolution because the Spatial path uses a fixed-size grid: as each frame produces more tokens, the fraction of historical KV those heads access shrinks (Table 2). On Self-Forcing the speedup rises from near 1× at standard resolution to 1.31× at 720P and 1.46× at 1080P; on LongLive, whose 30-second horizon further enlarges the cache, it reaches 1.62× and 1.95×.

Dummy Forcing scales more aggressively in raw FPS but degrades fidelity everywhere: on LongLive at 720P it trails HeadCast by ∼2.4 dB PSNR (12.83 vs. 15.22) at higher LPIPS (0.455 vs. 0.336), with a larger gap on Self-Forcing (18.81 vs. 25.87 dB)—the flickering of Figure 3, which HeadCast avoids while still delivering 1.62–1.95× and shrinking the steady-state KV cache by ∼33% on Self-Forcing (Appendix E).

PSNR/LPIPS are lower on long-video LongLive than on the 5-second backbones (15.22 vs. 25.87 dB at 720P)—a property of the reference-based metric, not a quality loss: over a 30-second rollout the sparsified model drifts further from the full-attention trajectory while *absolute* quality is preserved (VBench Total 79.05 vs. 79.07 at 720P, 77.26 vs. 76.84 at 1080P).

5.4 User Study

Since aggregate VBench is insensitive to eviction-induced flickering, we run a blind 2AFC study (protocol in Appendix H): shown two same-prompt videos, each participant makes one *overall-preference* choice, against full attention and against Dummy Forcing (Table 3).

The result is decisive against Dummy Forcing—HeadCast is preferred in 90.9% of comparisons, reflecting the flicker-free quality VBench misses. Against full attention the protocol forces a binary choice with no tie option, so a perceptually indistinguishable method would land at 50%; HeadCast receives 39.6% while running $1.62\times$ faster. We therefore do not claim perceptual parity—a majority can still tell the two apart when forced to pick—but nearly two in five prefer the accelerated output, far closer than existing training-free eviction.

5.5 Ablation Studies

Ablations use Self-Forcing on 5-second clips at 720P; the threshold and timestep sweeps confirming our defaults are in Appendix B.

Effectiveness of Distinct Head Archetypes. We ablate the two sparsity sources—Sink and Dummy heads as *Temporal Sparsity*, Spatial heads as *Spatial Sparsity*—on top of the All-Global baseline (Table 4). Spatial Sparsity is the dominant speed lever ($1.19\times$), fragmenting the dense attention map into independent local sub-matrices, but bears most of the fidelity cost (25.04 dB PSNR); Temporal Sparsity is gentler on both axes ($1.04\times$, 27.99 dB), merely capping the attended sequence length. Combining them reaches $1.31\times$ while VBench Total stays within 0.05 points of the All-Global baseline.

Thresholds Transfer Without Retuning. Must the thresholds be re-calibrated per model, resolution, or rollout length? They need not: every number in this paper—four backbones, three resolutions, and clip lengths from 5 to 30 seconds—uses the *same* $\theta{=}0.95$, $\theta_{\text{sc}}{=}0.755$ and the same 10×10 grid. They compare *cosine similarities* between a head's restricted and full outputs—scale-free quantities measuring whether a head's attention geometry is reproducible from a restricted context, rather than model-specific magnitudes. Both sweeps are smooth and monotone with the defaults at the Pareto knee, and re-classifying at a different AR step, timestep, or prompt reproduces the *identical* archetype for 90.3%, 85.8%, 79.8% of the 360 heads (Appendix B); the taxonomy thus tracks stable structure, not per-checkpoint calibration.

5.6 Iso-Throughput Comparison: the Trade-off Is Not Just a Budget Choice

Because Dummy Forcing runs slightly faster than our default configuration, one might suspect HeadCast's higher fidelity merely reflects a milder compression rate—a different point on the same curve, not a better curve. Sweeping the two thresholds moves heads between the sparse and Global paths, tracing our own frontier, plotted in Figure 4 against Dummy Forcing's single point.

Dummy Forcing lies far *below* the frontier rather than on it. Pushing HeadCast to 9.41 FPS—within 2% of Dummy Forcing's 9.60—still yields 25.32 dB PSNR against its 18.81 dB, and even our most aggressive setting (9.51 FPS, $\theta{=}0.91$) retains 24.51 dB. The advantage is therefore not a matter of compression rate: at comparable speed, routing heads by the four-way taxonomy preserves far more of the full-attention output than forcing them all into one aggressive mode. The frontier is also smooth and monotone, a continuous knob a fixed eviction policy cannot provide.

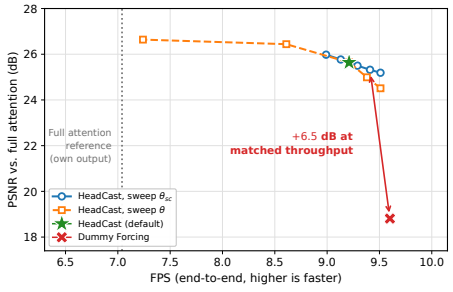


Figure 4: Speed/fidelity frontier (Self-Forcing, 5 s, 720P). At matched throughput (9.41 vs. 9.60 FPS) HeadCast attains 25.32 dB against Dummy Forcing's 18.81 dB. Both axes are end-to-end.

6 Conclusion

We presented HeadCast, a training-free framework that classifies each attention head once, at the maximum-noise step, into one of four archetypes—Sink, Dummy, Spatial, Global—and routes it to a matching path. Because the Spatial path uses a fixed-size grid, its savings grow with resolution ($1.62\times$ at 720P, $1.95\times$ at 1080P) while the retained Global heads preserve temporal consistency—making head-level heterogeneity a practical lever for efficient AR video generation.

REPRODUCIBILITY STATEMENT

All components needed to reproduce our results are documented in the paper and its appendix. The four head archetypes and the classification metric are specified in Section 4 (restricted-context proxies, the worst-case spatial score, and the mutually exclusive decision rules), and every hyperparameter we use—sliding-window and block sizes, the classification step S_{start}, the denoising timestep $t = 1000$, the thresholds $\theta_s = \theta_d = \theta_{\text{sp}} = 0.95$ and $\theta_{\text{sc}} = 0.755$, and the 10×10 spatial grid—is listed in the implementation details of Section 5; the same single setting is used for all four backbones and all resolutions. The evaluation protocol (VBench / VBench-Long prompt sets, prompt refinement, seed averaging, the full-attention reference used for PSNR/LPIPS, and single-GPU timing) is given in the same section. The appendix additionally reports the threshold and timestep sweeps, the classification-stability measurements, the head-archetype distribution and FLOPs analysis, the KV-cache memory accounting, the one-time classification overhead, the block-boundary discontinuity metric, and the full user-study protocol. HeadCast is training-free and applies to publicly released checkpoints of Self-Forcing, LongLive, Causal Forcing, and Reward Forcing; an anonymized implementation of the head classifier, the heterogeneous cache, and the head-specific attention kernels is provided as supplementary material.

ETHICS STATEMENT

This work accelerates inference for existing, publicly released autoregressive video generation models and introduces no new training data or model capability. All prompts come from established public benchmarks (VBench and the MovieGen prompt set), and no personal or sensitive data is used. The user study involved only anonymous, voluntary pairwise preference judgements over generated video clips, collected without any personally identifying information. As an efficiency method, HeadCast inherits the societal risks of the underlying generative models—including the potential for misleading synthetic video—but does not amplify them; it reduces the compute and energy required per generated video. We have read and adhere to the ICLR Code of Ethics.

AI USE STATEMENT

In this work, we used generative AI tools to assist with editing the manuscript for readability and with writing and refactoring parts of the implementation and evaluation code. We did not use generative AI tools to develop the conceptual framework or the head-archetype hypothesis, to formulate or prove mathematical claims, to design the methodology or experiments, to generate synthetic data, or to interpret the experimental results; all such contributions are the authors' own. Tasks such as survey/interview design and transcription of research recordings are not applicable to this work. We have reviewed all AI-assisted work: AI-assisted code was read, tested, and verified against the full-attention reference implementation by the authors, and all AI-edited text was checked for factual accuracy against our experimental logs. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets, 2023.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Pengtao Chen, Xianfang Zeng, Maosen Zhao, Mingzhu Shen, Wenjiang Cheng, Gang Yu, and Tao Chen. Sparse-vDiT: Unleashing the power of sparse attention to accelerate video diffusion transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 2957–2965, 2026.

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, 2025.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model tells you what to discard: Adaptive KV cache compression for LLMs. In *International Conference on Learning Representations (ICLR)*, 2024.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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

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

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

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. HunyuanVideo: A systematic framework for large video generative models, 2024.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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 (ICLR)*, 2026.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

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

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Adam Polyak et al. Movie gen: A cast of media foundation models, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022.

Sand AI. MAGI-1: Autoregressive video generation at scale, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2023.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Wan Team. Wan: Open and advanced large-scale video generative models, 2025.

Haocheng Xi, Shuo Yang, Yilong Zhao, Chenfeng Xu, Muyang Li, Xiuyu Li, Yujun Lin, Han Cai, Jintao Zhang, Dacheng Li, Jianfei Chen, Ion Stoica, Kurt Keutzer, and Song Han. Sparse VideoGen: Accelerating video diffusion transformers with spatial-temporal sparsity. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024.

Guangxuan Xiao, Jiaming Tang, Jingwei Zuo, Junxian Guo, Shang Yang, Haotian Tang, Yao Fu, and Song Han. DuoAttention: Efficient long-context LLM inference with retrieval and streaming heads. In *International Conference on Learning Representations (ICLR)*, 2025.

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

An Yang et al. Qwen2.5 technical report, 2024.

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 (ICLR)*, 2026.

Shuo Yang, Haocheng Xi, Yilong Zhao, Muyang Li, Jintao Zhang, Han Cai, Yujun Lin, Xiuyu Li, Chenfeng Xu, Kelly Peng, Jianfei Chen, Song Han, Kurt Keutzer, and Ion Stoica. Sparse VideoGen2: Accelerate video generation with sparse attention via semantic-aware permutation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a.

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

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 *Proceedings of the International Conference on Machine Learning (ICML)*, 2026.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

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

Peiyuan Zhang, Yongqi Chen, Haofeng Huang, Will Lin, Zhengzhong Liu, Ion Stoica, Eric P. Xing, and Hao Zhang. Faster video diffusion with trainable sparse attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a.

Peiyuan Zhang, Yongqi Chen, Runlong Su, Hangliang Ding, Ion Stoica, Zhengzhong Liu, and Hao Zhang. Fast video generation with sliding tile attention. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2025b.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Hongzhou Zhu, Min Zhao, Guande He, Hang Su, Chongxuan Li, and Jun Zhu. Causal forcing: Autoregressive diffusion distillation done right for high-quality real-time video generation. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2026.

In this appendix, we provide additional details and results, including:

- Per-head attention-pattern visualizations (Appendix A).
- Additional ablation studies—classification thresholds, the denoising timestep, and the stability of the one-shot classification (Appendix B).
- An illustration of the Spatial Path attention pattern (Appendix C).
- The full head-archetype distribution and a theoretical FLOPs analysis (Appendix D).
- The KV-cache memory cost of each method (Appendix E).
- The one-time classification overhead (Appendix F).
- A block-boundary discontinuity metric and further inter-frame flicker comparisons (Appendix G).
- The user study protocol (Appendix H).
- Additional qualitative results on Self-Forcing (5 s) and LongLive (30 s) (Appendix I).

A Per-Head Attention Pattern Visualization

To complement the four head archetypes introduced in the main paper, we enlarge one representative head per archetype—each a different head from the examples in the main paper's Figure 1—in Figures 5 and 6. Each head occupies two rows (top: decoding frame 6; bottom: frame 18), with three columns: the *raw* attention map, its per-frame (*temporal*) aggregation, and its per-position (*spatial*) aggregation, all at the maximum-noise step ($t = 1000$). Comparing the two rows shows that each head keeps its archetype from frame 6 to frame 18—a visual counterpart to the quantitative stability in Figure 8 that licenses our *one-shot* classification at a single denoising step.

Sink heads (Figure 5, top) concentrate a disproportionate share of their attention mass on the *first* cached block, while the spatial view exhibits no coherent locality structure—the mass falls on a few scattered spatial positions—so these heads contribute little position-specific information to the current step. For the shown head (Layer 23, Head 2) the leading frame of this block alone receives 38% of the mass at decoding frame 6 and remains by far the dominant frame at decoding frame 18 (15%, roughly $3\times$ the uniform share).

Dummy heads (Figure 5, middle) collapse onto the *current* block: essentially all attention mass falls on its frames ($\sim$33% each), with each query attending almost exclusively to its own frame and negligible weight on older history, so these heads read only a constant-size tail of the cache and are among the cheapest to serve.

Spatial heads (Figure 5, bottom) do not lock onto a single frame; instead, the raw map reveals a banded, near-diagonal structure in which each query token attends to a fixed local neighborhood replicated across the cached frames. The per-position (spatial) view makes this locality explicit.

Global heads (Figure 6) spread their attention broadly across all cached frames and all spatial positions, with no single frame exceeding roughly one-fifth of the total mass. These heads carry the long-range temporal context that HeadCast explicitly preserves by routing them to the full-attention path.

B Additional Ablation Studies

We provide the ablations deferred from the main paper: the two classification thresholds and the denoising timestep t. Unless stated otherwise, all runs use Self-Forcing on 5-second clips at 720P.

Impact of Classification Thresholds. HeadCast is controlled by two thresholds: the cosine threshold $\theta := \theta_s = \theta_d = \theta_{\text{sp}}$ that gates Sink/Dummy/Spatial assignment, and the spatial score threshold θ_{sc}. We sweep each independently, measuring FPS and PSNR.

As shown in Figure 7, tightening θ from 0.91 to 0.99 raises PSNR from 24.51 to 26.64 dB at the price of dropping FPS from 9.51 to 7.24, while θ_{sc} controls a milder trade-off (± 0.52 FPS, ∓ 0.79 dB). Both curves are smooth and monotone, and our defaults sit at the Pareto knee of each. Because the thresholds compare *cosine similarities* of attention patterns—quantities that are normalized and largely architecture-agnostic—we apply this single set (θ=0.95, θ_{sc}=0.755) unchanged across all

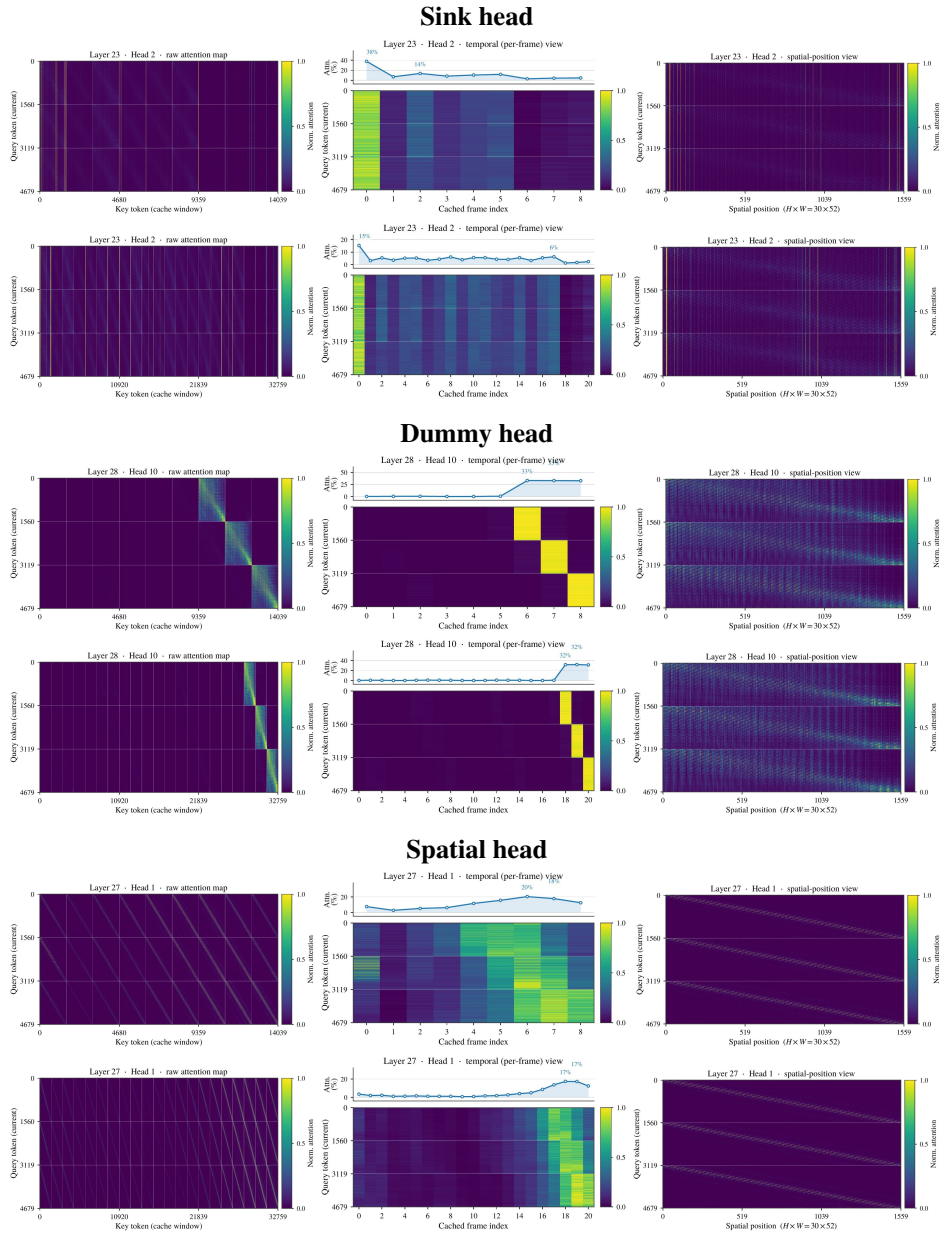


Figure 5: Sink, Dummy, and Spatial heads. One representative head per archetype, each occupying two rows (frame 6 then frame 18). Columns: raw map, per-frame (temporal) aggregation, per-position (spatial) aggregation. The *Sink* head (Layer 23, Head 2) concentrates on the first cached block; the *Dummy* head (Layer 28, Head 10) collapses onto the current block; the *Spatial* head (Layer 27, Head 1) attends to a fixed local neighborhood.

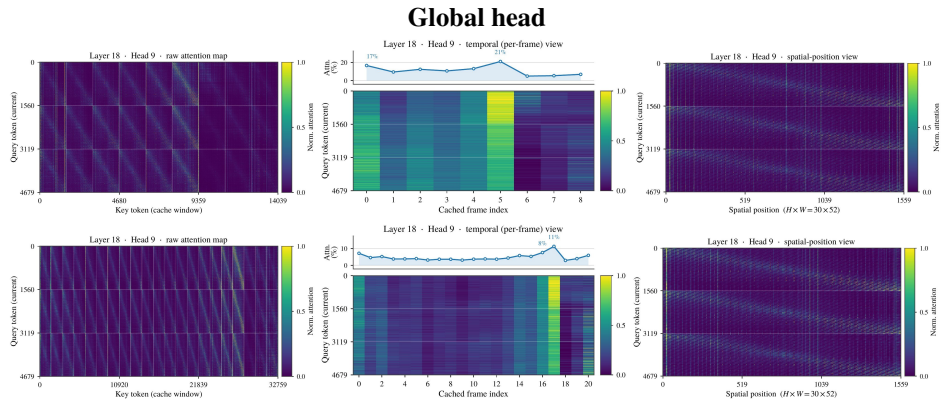


Figure 6: Global head. One representative head occupying two rows (frame 6 then frame 18). Columns: raw map, per-frame (temporal) aggregation, per-position (spatial) aggregation. The *Global* head (Layer 18, Head 9) spreads its attention broadly across the whole history.

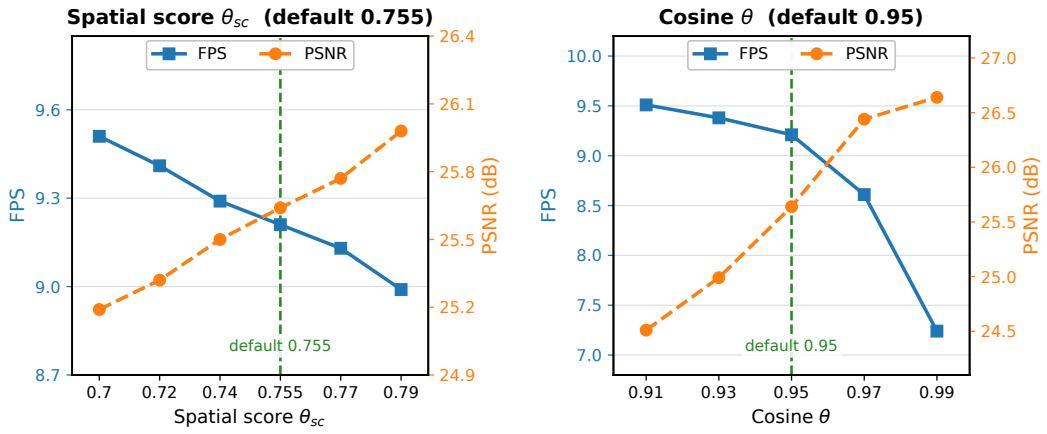


Figure 7: Ablation of classification thresholds (Self-Forcing, 5 s, 720P). *Left:* spatial score threshold θ_{sc}. *Right:* cosine threshold θ. Lowering either threshold routes more heads to sparse paths, trading PSNR for FPS; green dashed lines mark our defaults (θ_{sc}=0.755, θ=0.95).

four backbones and all resolutions, with no per-model retuning, indicating that the classification is governed by stable structural geometry rather than dataset- or model-specific calibration.

Impact of the Classification Timestep. We examine the denoising timestep t at which the one-shot classification is performed (Table 5).

t	Sink	Dummy	Spatial	Global	PSNR↑	LPIPS↓
1000	29.4	**136.3**	84.1	**110.2**	**25.87**	**0.0568**
750	41.3	149.4	81.7	87.7	25.19	0.0612
500	48.1	166.3	69.3	76.2	24.79	0.0649
250	53.0	190.8	50.4	65.9	24.52	0.0674

Table 5: Ablation on the classification denoising timestep t (Self-Forcing, 5 s, 720P). As t decreases, low-noise denoising focuses on local detail, so more heads attend only to their current frame and are classified as *Dummy* (136.3 $\rightarrow$ 190.8), while *Global* heads collapse (110.2 $\rightarrow$ 65.9); frame fidelity degrades accordingly, and $t = 1000$ gives the best PSNR/LPIPS. Head counts are out of 360.

As denoising proceeds from $t = 1000$ toward lower noise levels, the model concentrates on local, high-frequency detail, so progressively more heads attend only to their own current frame—the defining behavior of the *Dummy* archetype, which our decision hierarchy assigns with priority. The Dummy count therefore grows steadily as t falls (from 136.3 at t=1000 to 190.8 at t=250), with a smaller rise in Sink heads, while the number of *Global* heads—those carrying the long-range temporal context—collapses (110.2 $\rightarrow$ 65.9). Heads that genuinely require global history are thereby

misrouted to the constant-size Dummy/Sink window and lose the keys needed for temporal consistency, introducing inter-frame flickering; both PSNR and LPIPS degrade monotonically as the classification timestep drops from 1000 to 250. Classifying at the maximum noise level ($t = 1000$), where the most heads still expose their full long-range dependency structure, therefore yields the best fidelity and supports our default choice.

Stability of the One-Shot Classification. HeadCast classifies each head exactly once and reuses that assignment for the entire rollout, so we verify that the assignment is robust to the conditions under which it is computed: the autoregressive (AR) step, the denoising timestep, and the text prompt. For each axis we re-run the classification along it and measure the fraction of the 360 heads that receive the *identical* four-way archetype (Figure 8). The assignment is highly consistent on every axis—90.3% across AR steps (frame 12 vs. 18, two blocks apart), 85.8% across denoising timesteps, and 79.8% across text prompts—confirming Observation 3 in the main paper: a single classification at a representative AR step and the maximum-noise timestep suffices, and re-profiling at every step is unnecessary.

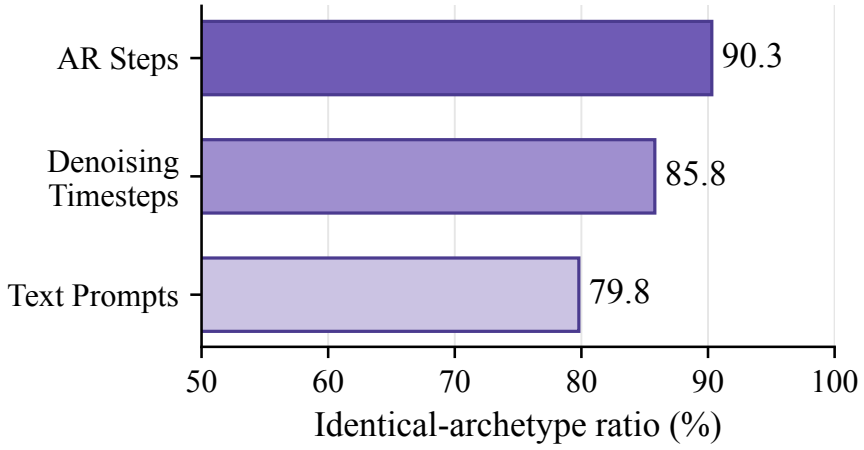


Figure 8: Stability of the one-shot classification (Self-Forcing, 5 s, 720P). Each bar is the fraction of the 360 heads that keep the *identical* four-way archetype when the classification is repeated across that axis (mean over all pairs of sampled points).

C Spatial Path Attention Pattern

Figure 9 illustrates the block-to-block attention pattern of the Spatial Path described in Section 4. Each frame is partitioned into non-overlapping spatial grids, and a Spatial head's query attends only to the historical key-value tokens lying in its own grid, so the dense global attention matrix decomposes into many small independent sub-matrices that batch onto FlashAttention.

D Head Archetype Distribution and Theoretical Compute Savings

We report the full four-way head distribution under the default configuration and use it to attribute HeadCast's speedup to its constituent archetypes. Unlike the merged `Temp./Spat.` columns of the head-archetype ablation in the main paper (which fold Sink and Dummy into a single *Temporal* group), Table 6 gives the complete Sink/Dummy/Spatial/Global counts at the default t=1000 for LongLive across resolutions—the running example for the FLOPs analysis below; all $30 \times 12 = 360$ heads are accounted for.

Theoretical compute savings. The attention FLOPs of a single head are dominated by its two matmuls—$QK^\top$ and softmax V—totalling $4\,B\,L_q\,L_k\,d_{\text{head}}$. Because every comparison below is a ratio, this constant factor cancels, so we drop it and write the cost as $B\,L_q\,L_k\,d_{\text{head}}$. Under our inference configuration each block has sequence length L, so a query block has $L_q = L$ and attends to a window of $L_k = WL$ tokens. Taking LongLive's 30-second sliding window as a concrete

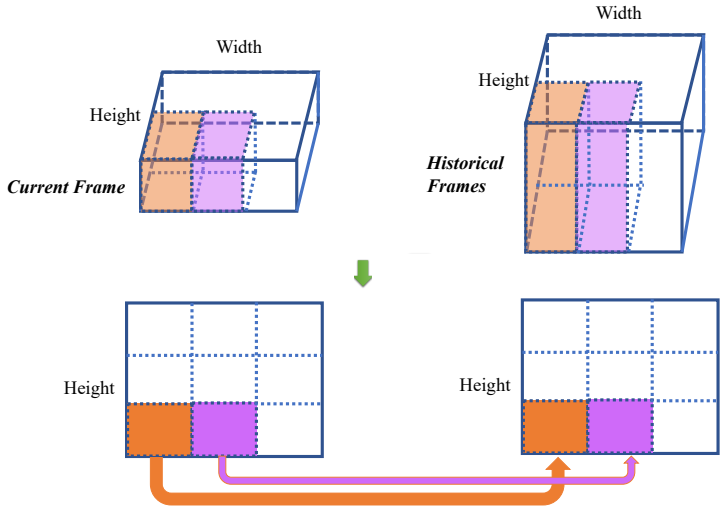


Figure 9: Illustration of the attention computation in the Spatial Path. The spatial resolution of the frames is partitioned into non-overlapping grids. Each query attends only to historical key-value tokens within its own spatial grid (block-to-block), decomposing the dense global attention into independent sub-matrices.

Resolution	Sink	Dummy	Spatial	Global
480P	1.6	93.8	168.2	96.5
720P	1.9	98.2	167.2	92.7
1080P	1.5	91.7	203.2	63.7

Table 6: Head archetype distribution (LongLive, default t=1000). Counts are out of 360 heads (30 layers $\times$ 12 heads); entries are means and may not sum to exactly 360 due to rounding.

example (W=8: 7 cached blocks plus the current block), a full-attention (Global) head therefore costs $8BL^2d$. Each archetype reduces this differently:

- **Sink** attends only to the first (sink) block and the current block: $8BL^2d \to 2BL^2d$ (75% reduction).
- **Dummy** attends only to the previous block and the current block: $8BL^2d \to 2BL^2d$ (75% reduction).
- **Spatial** restricts each query to its own fixed 10×10 cell, which at 480P (a 30×52 latent grid) covers $\approx \frac{1}{16}$ of the frame: $8BL^2d \to \frac{1}{2}BL^2d$ ($\approx$94% reduction at 480P, and more at higher resolution, where the fixed-size cell spans a smaller fraction).
- **Global** is unchanged: $8BL^2d$ (0% reduction).

Integrating over the measured distribution, the attention-FLOPs ratio is $\frac{2N_s+2N_d+\frac{1}{2}N_{sp}+8N_g}{8\,N_{total}}$, giving a $\sim$64% reduction at 480P that grows at higher resolution, as the fixed-size Spatial cells cover a smaller fraction of each frame. This exceeds the measured end-to-end speedup (1.62–1.95$\times$, i.e. a 38–49% wall-clock reduction). The gap to the theoretical FLOPs saving has three sources: attention is only part of the per-block compute (the diffusion MLP and projection layers are unaffected); HeadCast dispatches its archetype groups through separate attention kernels—about three launches per block instead of one—so the added kernel-launch overhead offsets part of the FLOPs reduction; and the wall-clock figure additionally absorbs the full-context warm-up and the one-time classification, neither of which is accelerated.

E KV-Cache Memory Cost

A KV-cache compression method should also shrink memory, so we report the steady-state KV-cache size for the full-attention baseline, Dummy Forcing, and HeadCast (Figure 10). HeadCast consistently caches less than full attention—about 33% less on Self-Forcing—by capping the history that Sink and Dummy heads retain, while still keeping substantially more context than Dummy

Forcing's aggressive eviction, which is precisely what spares it the inter-frame flicker that eviction produces.

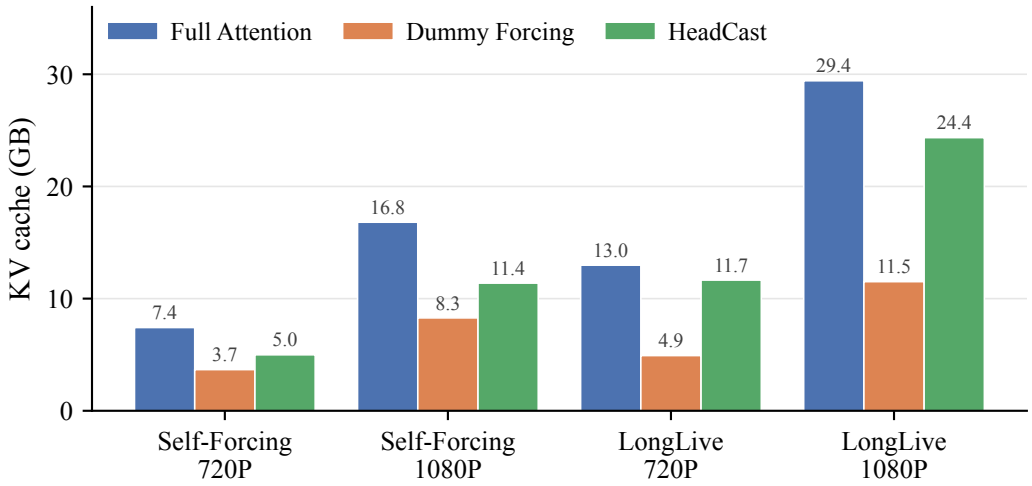


Figure 10: Steady-state KV-cache size (GB) for the full-attention baseline, Dummy Forcing, and HeadCast. HeadCast caches less than full attention at every setting (about 33% less on Self-Forcing) while retaining substantially more history than Dummy Forcing.

Theoretical KV-cache reduction. With a W-block window (W=8), a full head caches WL tokens. HeadCast retains KV per archetype: Sink and Dummy heads keep only $2L$ tokens (their anchor block plus the current block, a 75% reduction), while Spatial and Global heads still require the full window (Spatial partitions only the spatial dimensions, dropping no frames). The expected steady-state KV reduction is therefore $\frac{N_s+N_d}{N_{\text{total}}} \times 0.75$. For the Self-Forcing distribution ($N_s+N_d \approx 166/360 \approx 46\%$) this predicts a $\sim$ 34% reduction, close to the measured $\sim$ 33% in Figure 10. LongLive, whose Sink+Dummy share is smaller, shows a correspondingly milder reduction.

F Classification Overhead

HeadCast's only method-specific extra cost is the one-time classification, which fires on a single autoregressive block. **This cost is already included in every FPS and speedup we report**—all timings are end-to-end wall-clock measurements over the complete generation—so the numbers in the main paper are true end-to-end figures and require no further discounting. Table 7 instead quantifies how large a share of that end-to-end budget the classification occupies. It is small and, crucially, *amortizes with the rollout length*: on 5-second Self-Forcing clips it accounts for 5–8% of the budget, but on the flagship 30-second LongLive setting it drops below 1.5%—only 0.6% at 1080P. This amortization is one reason the end-to-end gains are largest exactly where the method is aimed: the 1.62× and 1.95× reported for 30-second LongLive at 720P and 1080P are obtained with the classification cost fully absorbed.

Backbone (length)	**480P**	**720P**	**1080P**
Self-Forcing (5 s)	7.9%	7.1%	4.8%
LongLive (30 s)	1.3%	1.0%	**0.6%**

Table 7: Share of the end-to-end generation budget taken by the one-time classification. This cost is *already included* in all reported FPS/speedup numbers; the table shows how large it is, not a correction to be applied. The share amortizes as the video lengthens, becoming negligible on the 30-second setting.

G Inter-Frame Flickering

A block-boundary discontinuity metric. Eviction-induced flicker is *localized*: it appears as an abrupt content change exactly at the autoregressive block boundaries, where the sliding cache is up-

dated and stale keys are dropped. We measure it directly. For consecutive decoded frames we compute the mean absolute pixel difference $d_t = \frac{1}{HWC}\sum|f_t - f_{t-1}|$, and define the *block-boundary discontinuity* (BBD) as the ratio of the average d_t over block-boundary frames to the average d_t within blocks. A temporally smooth clip has $\mathrm{BBD} \approx 1$; values above 1 mean the boundaries jump more than ordinary motion—the pop-in/pop-out artifact of Figure 3. Because the ratio is localized at boundaries, it isolates exactly the failure mode that frame-averaged scores such as VBench dilute away.

Setting	Full	HeadCast	Dummy	Δ_{Dummy}
Self-Forcing (5s), 480P	1.192	1.192	1.323	$+0.131$
Self-Forcing (5s), 720P	1.235	1.238	1.272	$+0.037$
Self-Forcing (5s), 1080P	1.265	1.250	1.285	$+0.020$
LongLive (30s), 480P	1.304	1.260	1.777	$+0.473$
LongLive (30s), 720P	1.245	1.242	1.536	$+0.291$
LongLive (30s), 1080P	1.232	1.166	1.395	$+0.163$

Table 8: Block-boundary discontinuity (BBD; 1.0 = no boundary jump, lower is smoother). HeadCast matches the full-attention baseline at every setting, whereas Dummy Forcing's eviction inflates the discontinuity—most severely on the 30-second LongLive clips. Δ_{Dummy} is Dummy minus Full, computed at full precision (so it may differ from the difference of the rounded entries by 0.001).

Table 8 isolates the effect of cache eviction by comparing each accelerated method against full attention. HeadCast adds essentially *no* boundary discontinuity: its BBD equals the full-attention baseline on the 5-second clips and is even slightly *lower* on the 30-second clips (HeadCast $-$ Full ≤ 0 throughout). Dummy Forcing instead inflates it, and the inflation scales with the rollout length—which is exactly what one expects of an artifact that accumulates as the cache is repeatedly evicted. On 5-second clips, where eviction happens only a handful of times, the excess is small ($+0.02$ to $+0.13$); over the 30-second horizon it compounds to $+0.16$–$+0.47$ (13–36% relative), peaking on the flagship long-video LongLive setting. Flicker is thus fundamentally a long-rollout problem, and it is precisely there that HeadCast's explicit Global heads pay off. This trend mirrors the qualitative examples below and the PSNR/LPIPS gap in the main paper—an artifact the metric and figures expose but that aggregate VBench dilutes away.

To complement the single example in the main paper (Figure 3), we provide two further side-by-side comparisons that isolate the inter-frame flickering induced by aggressive cache eviction. Each example spans a five-frame window centered on an autoregressive block boundary, where Dummy Forcing's coarse eviction is most prone to hallucinating or dropping structure. As before, all three methods (Dummy Forcing, Full Attention, and our HeadCast) are run from the same prompt on LongLive at 720P, and the frame index is shared across rows so that a given column shows the same timestamp under each method. The red box marks the region of interest on the Dummy Forcing row.

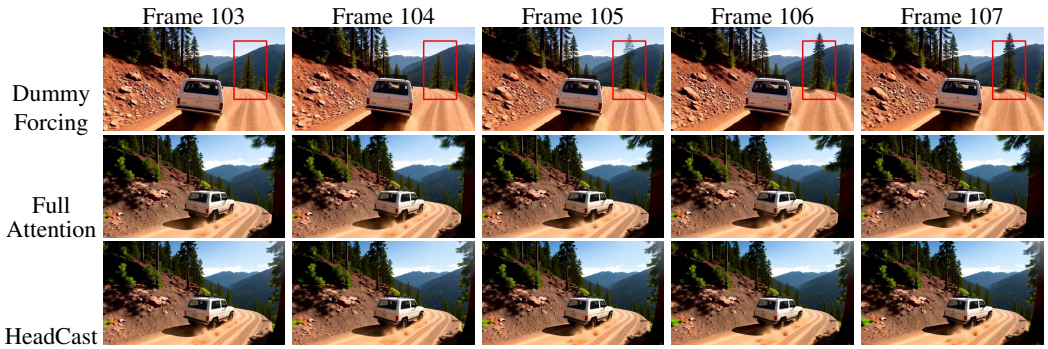
In both examples, Full Attention and HeadCast keep the highlighted structure temporally stable across the boundary, whereas Dummy Forcing introduces an abrupt, frame-to-frame change—precisely the artifact that lowers its PSNR/LPIPS yet, once VBench's temporal sub-dimensions are pooled into the aggregate score, is largely washed out.

H User Study Protocol

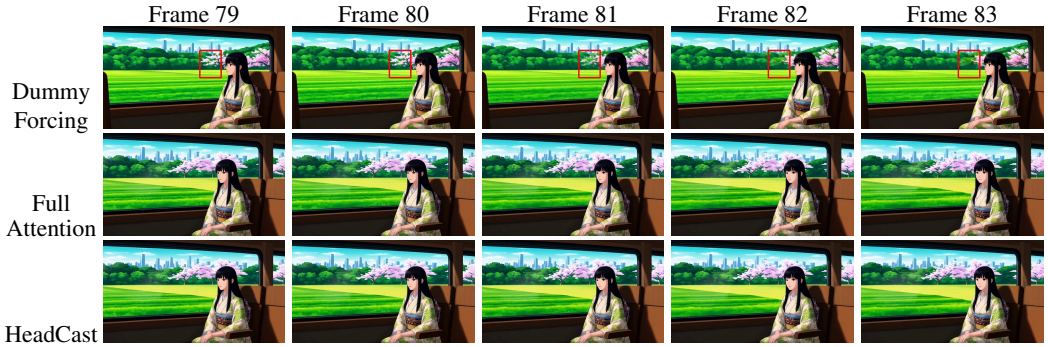
To complement the automatic metrics with human judgment, we run a user study based on the Two-Alternative Forced Choice (2AFC) protocol. In each question a participant is shown two videos generated from the *same* prompt and answers a single question—*which video do you prefer overall?*—picking the one that looks more realistic and natural, with fewer visible artifacts. We deliberately collect one holistic preference rather than splitting the judgment across sub-dimensions: the artifacts our method targets—inter-frame flicker and structural drift at block boundaries—degrade the overall viewing experience rather than any single attribute, so a forced overall choice is the most direct measure of which video a viewer would rather watch.

We evaluate HeadCast in two head-to-head settings:

- **HeadCast vs. Full Attention**, which tests whether our acceleration preserves the quality of the unmodified model; and



(a) Mountain-road scene: a pine tree on the right (red box) abruptly pops in and out across the block boundary at frame 105 under Dummy Forcing.



(b) Train-window scene: cherry blossoms seen through the window (red box) flicker across the block boundary at frame 81 under Dummy Forcing.

Figure 11: Additional inter-frame flickering examples at AR block boundaries (LongLive, 30 s, 720P). Each five-frame window is centered on a block edge. Dummy Forcing's aggressive cache eviction introduces an abrupt, frame-to-frame change in the boxed region, whereas *Full Attention* and our *HeadCast* keep the scene temporally consistent.

- **HeadCast vs. Dummy Forcing**, the strongest training-free baseline, which tests whether our head-specific eviction yields visibly better video than aggressive eviction.

For each setting we generate 160 same-prompt pairs spanning both clip lengths—128 short (5 s) and 32 long (30 s)—with each pair sharing prompt and seed so the only difference is the attention pathway. The prompts are sampled uniformly at random from the Movie Gen prompt suite. We then split the 160 pairs into 10 groups by randomly assigning pairs within each length stratum, so the grouping is unbiased while every group still holds a comparable mix ($\sim$12–13 short and 3–4 long clips). Every group is scored by two *different* annotators in two independent rounds, giving two votes per pair ($\sim$320 per setting) over 20 participants in total.

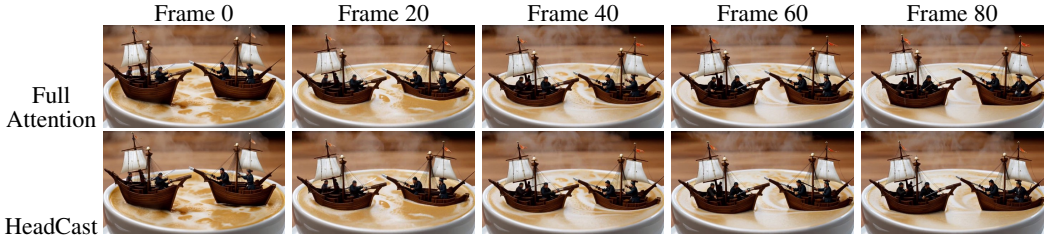
Within each pair the two videos are shown as *video 1*/*video 2* in randomized order, with the slot-to-method mapping withheld and stored separately, so the study is fully double-blind; each question is a strict two-way forced choice with no "tie" option. We report the percentage of votes preferring HeadCast over its opponent (above 50% favors HeadCast); the results are summarized in the main paper.

I Additional Qualitative Results

We compare HeadCast with the full-attention model frame by frame: in each block the top row is full attention and the bottom row is HeadCast, with five frames sampled at equal intervals. All prompts are drawn from the MovieGen prompt set (Polyak et al., 2024)—the same suite used for VBench-Long evaluation in the main paper. On 5-second Self-Forcing (720P; Figures 12 and 13)—a neon-lit Tokyo street, ships in a coffee cup, puppies in the snow—the two rows are nearly identical despite

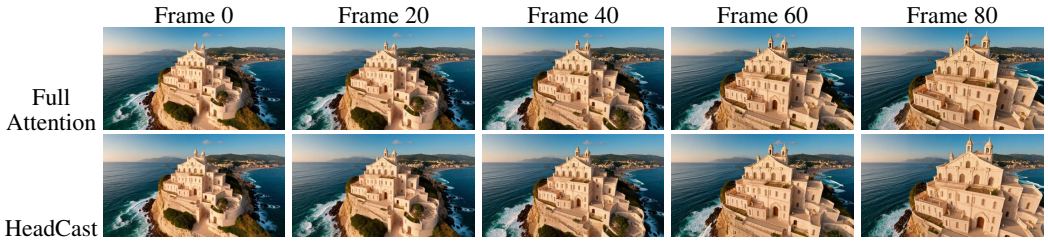


(a) A stylish woman walking down a neon-lit Tokyo street.

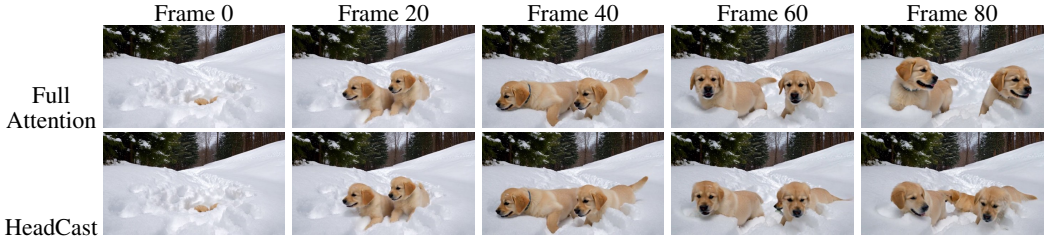


(b) Two pirate ships battling inside a cup of coffee.

Figure 12: Additional qualitative results on Self-Forcing 5s (720P), part 1. Rows top to bottom: Full Attention, HeadCast. Five frames are sampled uniformly across the 5-second clip.



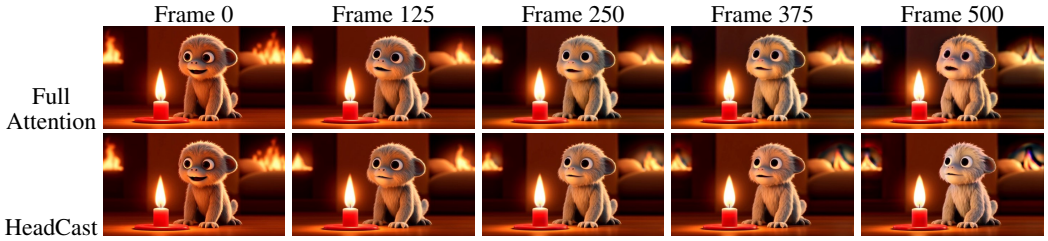
(a) A drone shot circling a historic church on the Amalfi Coast.



(b) Golden retriever puppies playing in the snow.

Figure 13: Additional qualitative results on Self-Forcing 5s (720P), part 2. Rows top to bottom: Full Attention, HeadCast. Five frames are sampled uniformly across the 5-second clip.

HeadCast's much smaller KV cache. The longer 30-second LongLive setting (720P; Figures 14 and 15) is a harder test, yet HeadCast still matches the baseline across the whole rollout.

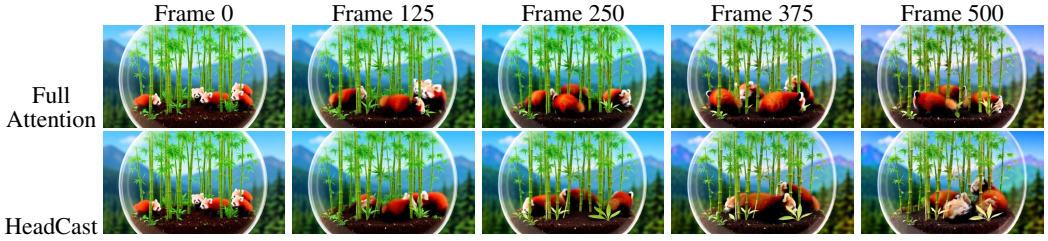


(a) A meerkat figurine beside a lit candle.



(b) A miniature figure inside a glass dome.

Figure 14: Additional qualitative results on LongLive 30s (720P), part 1. Rows top to bottom: Full Attention, HeadCast. Five frames are sampled uniformly across the 30-second clip.



(a) Red pandas in a bamboo terrarium.



(b) An elderly man walking along a city street.

Figure 15: Additional qualitative results on LongLive 30s (720P), part 2. Rows top to bottom: Full Attention, HeadCast. Five frames are sampled uniformly across the 30-second clip.