ROLLING SINK: BRIDGING LIMITED-HORIZON TRAINING AND OPEN-ENDED TESTING IN AUTOREGRESSIVE VIDEO DIFFUSION

Anonymous authors
Paper under double-blind review

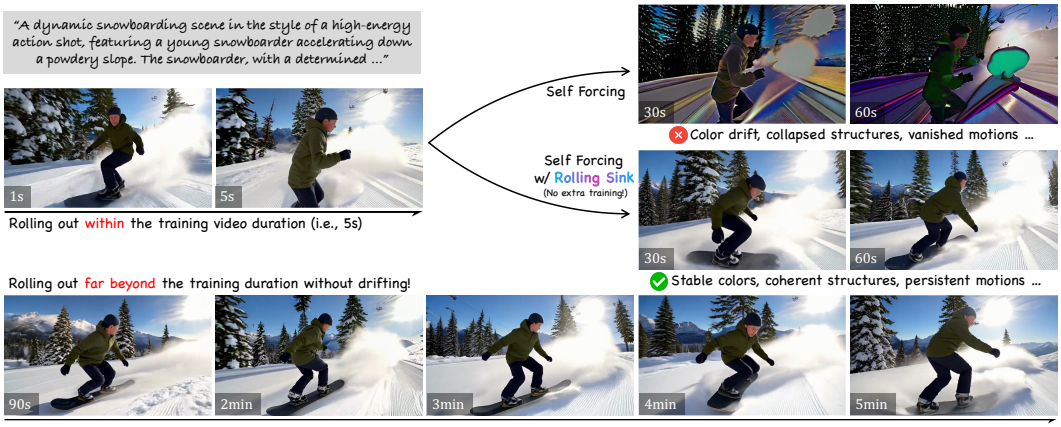

Figure 1: **Rolling Sink extends a short-horizon-trained AR video diffusion model to much longer test-time rollouts.** Without extra training, it keeps a 5s-trained model rolling far beyond its training horizon under a bounded cache: a 5-minute rollout here, and a 30-minute qualitative example in Fig. S1 (App. A).

ABSTRACT

Autoregressive (AR) video diffusion models in the Self Forcing family generate a video chunk by chunk, each conditioned on the ones before it, so they can keep rolling out with no preset length. In practice they are trained on short, fixed clips (*e.g.*, 5s). Once a rollout passes this training horizon, the cache fills with self-generated chunks the model never saw in training, and quality degrades: colors oversaturate, identities drift, structures collapse, and motion fades. We frame this failure as a *cache-behavior mismatch*, a domain shift between the within-horizon caches seen in training and those produced by open-ended testing, and address it with **Rolling Sink**, a training-free[1] inference-time cache policy. Within the base model's fixed cache budget, Rolling Sink holds the cache close to its stable within-horizon behavior: it restores low-drift content, sliding temporal indices, and evolving semantics, and keeps a small recent context. Attention sink and temporal re-indexing are existing mechanisms. Our contribution is the mismatch formulation, the joint bounded-cache policy, and the Rolling Semantics schedule. On two 5s-trained bases, Self Forcing and Causal Forcing, Rolling Sink attains the best averaged rank on `VBench-Long` at both 1- and 5-minute rollouts under the same inference setting, at about 0.56% throughput change, and holds GPU memory constant. It also sustains a 30-minute qualitative rollout, $360\times$ its training horizon. We validate at 1- and 5-minute rollouts and do not claim guaranteed quality at arbitrary duration.

[1]Training-free means no additional training on top of the base AR model, which is itself trained. Because open-ended testing can exceed any finite training duration, this gap cannot be fully closed by training on longer clips alone, which motivates an inference-time solution.

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.

1 INTRODUCTION

Shot length in film is open-ended: a take may last seconds or run for minutes, like the 16.5-minute opening dialogue of Steve McQueen's *Hunger* (McQueen, 2008) or the minute-long tricycle shot in Kubrick's *The Shining* (Kubrick, 1980). This motivates open-ended generation, where the length is not fixed and the model must keep producing coherent frames for as long as asked.

Large video diffusion models (OpenAI, 2025; Wan et al., 2025; Kling, 2025; Google, 2025; Runway, 2025; Seedance et al., 2026) denoise all frames at once, which ties them to short, fixed clips and rules out this setting. AR video diffusion models in the Self Forcing family (Huang et al., 2025a; Zhu et al., 2026; Yang et al., 2025; Liu et al., 2025) instead predict each chunk from previously generated ones, which lets them keep rolling out with no preset length. These models are trained on short clips (*e.g.*, 5s). Within the training horizon they are stable, but once a rollout passes it they degrade fast: oversaturated colors, collapsed structures, inconsistent identities, and vanishing motion (Fig. 1).

We read this degradation as a domain shift on the AR cache. The prompt embedding is fixed, and each chunk starts from the same noise distribution. The only input that changes across the rollout is thus the cache, *i.e.*, past self-generated chunks. Within the horizon, the cache stays in the training distribution, so the model behaves as trained. Beyond the horizon, the cache falls outside it. Each prediction is then biased, and the errors compound over the rollout, an instance of exposure bias (Bengio et al., 2015; Lamb et al., 2016). Limited-horizon training cannot cover open-ended lengths, leaving the cache as the only test-time lever. We call this gap a *cache-behavior mismatch* and reduce it by making the test-time cache resemble the within-horizon cache seen in training, along the three factors the model conditions on: low-drift content, sliding temporal indices, and evolving semantics.

Building on this view, we propose **Rolling Sink**, a training-free[1] inference-time cache policy that runs under the base model's fixed budget and few-step denoising, keeping the base's streaming efficiency (App. B). To restore content, it pins a few *sink* chunks generated within the training horizon, following the attention-sink idea from streaming language models (Xiao et al., 2023), and re-indexes them to the sliding window before the current chunk, as in prior horizon-extension work (Yi et al., 2025; Yesiltepe et al., 2025). To restore evolving semantics, *Rolling Semantics* traverses the finite bank of low-drift chunks in a forward-reverse schedule and reads a sliding window from it, while one retained recent chunk anchors local motion.

Attention sinks and temporal re-indexing are existing mechanisms, so we do not claim them as contributions. Rolling Sink contributes the framing and the policy that combines them: (i) casting the horizon gap as a cache-behavior mismatch, (ii) a bounded-cache policy that jointly maintains low-drift content, sliding temporal indices, and evolving semantics, and (iii) the Rolling Semantics schedule over a finite low-drift bank. This sets Rolling Sink apart from prior methods that mainly target a single axis: positional extrapolation (Yesiltepe et al., 2025; Li et al., 2026) or context retention (Yi et al., 2025; Kim et al., 2026). Tab. 1 gives the per-method comparison.

We evaluate on `VBench-Long` (Huang et al., 2024; 2025b; Zheng et al., 2025) over 1- and 5-minute rollouts, with qualitative rollouts in Fig. 3 and a 30-minute example in App. A. On two 5s-trained bases, Self Forcing (Huang et al., 2025a) and Causal Forcing (Zhu et al., 2026), Rolling Sink attains the best averaged rank at both lengths under the same inference setting, and its Self Forcing variant ranks first even against training-based baselines, including LongLive (Yang et al., 2025) trained on 1-minute clips. Averaged rank aggregates over all `VBench-Long` dimensions, so Rolling Sink does not lead on every one. Sec. 4 reports the per-dimension trade-offs. Rolling Sink keeps generating at constant GPU memory. The 1- and 5-minute numbers are quantitative, the 30-minute result is qualitative, and we do not claim guaranteed quality at arbitrary duration.

In summary, our contributions are: ① **Formulation.** We frame long-horizon AR drift as a cache-behavior mismatch between limited-horizon training and open-ended testing, and identify bounded-cache maintenance as the lever for narrowing it. ② **Method.** We introduce Rolling Sink, a training-free bounded-cache policy that jointly maintains low-drift content, sliding temporal indices, and evolving semantics, with Rolling Semantics as its one new component. ③ **Evidence.** Rolling Sink attains the best averaged rank on `VBench-Long` at 1- and 5-minute rollouts on two 5s-trained bases, at about 0.56% throughput change, and sustains a 30-minute qualitative rollout.

2 Related Work

2.1 Autoregressive Video Diffusion

Large video diffusion models (Blattmann et al., 2023; Yang et al., 2024; Kong et al., 2024; OpenAI, 2025; Wan et al., 2025; Google, 2025; Runway, 2025; Seedance et al., 2026; Zhang et al., 2026) build on image diffusion (Ho et al., 2020; Rombach et al., 2022; Song et al., 2020; He et al., 2024a;b; 2025) and denoise all frames jointly with bidirectional attention (Peebles & Xie, 2023), which ties them to short, fixed clips. AR video diffusion instead factorizes a video into chunk-wise conditionals and predicts each chunk from prior ones (Chen et al., 2024; Deng et al., 2024; Yin et al., 2024b; 2025; Teng et al., 2025; Chen et al., 2025; Guo et al., 2025; Zhang et al., 2025a; Ren et al., 2025; Zhou et al., 2026), in principle enabling unbounded rollout. Self Forcing (Huang et al., 2025a) aligns the within-horizon cache by training on the model's own generations, and Causal Forcing (Zhu et al., 2026), LongLive (Yang et al., 2025), and Rolling Forcing (Liu et al., 2025) extend the recipe with better distillation or longer rollouts. All of these modify training yet still train on short clips (5–6.5s, up to 1 minute for LongLive), so they drift once the rollout exceeds the trained length (Fig. 3). Rolling Sink leaves training untouched and controls the cache at inference.

2.2 Training-Free Test-Time Horizon Extension

A second line keeps the base model frozen and edits the cache at test time. One group corrects generation dynamics: ∞-RoPE (Yesiltepe et al., 2025) and FLEX (Li et al., 2026) extend temporal positional indices past the trained range, and Pathwise TTC (Xiang et al., 2026) corrects sampling trajectories against a stable reference. These target positional extrapolation and do not restore low-drift content once recent chunks drift. Another group manages historical context under bounded memory. Deep Forcing (Yi et al., 2025) keeps persistent deep-sink chunks and realigns their RoPE phases to the current timeline, MemRoPE (Kim et al., 2026) pairs memory tokens with online RoPE indexing, and Head Forcing (Tian et al., 2026), Echo Forcing (Wu et al., 2026), and Pack Forcing (Mao et al., 2026) adopt head-wise, scene-level, or hierarchical cache policies. These retain a low-drift or memory prefix but pin its semantics, so the retained context stops evolving with the rollout. The attention-sink and RoPE re-indexing mechanisms that Rolling Sink reuses originate here and in streaming language models (Xiao et al., 2023), which keep initial sink tokens beside a recent window under relative positions. Rolling Sink instead matches the cache to its within-horizon behavior along three factors at once: low-drift content, sliding temporal indices, and evolving semantics. Among the compared methods, it is the only training-free policy to maintain all three under the base model's bounded budget. Its new component, Rolling Semantics, keeps the retained low-drift context evolving in semantics, the axis the retention methods above leave fixed. Tab. 1 summarizes where each method sits.

3 Methodology

3.1 Preliminaries

An AR video diffusion model generates a video as a sequence of N chunks $\boldsymbol{x}_{[0,N)} = (\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{N-1})$, denoising each from the prior ones:

$$p_\theta\left(\boldsymbol{x}_{[0,N)}\right) = \prod_{i=0}^{N-1} p_\theta(\boldsymbol{x}_i \mid \boldsymbol{x}_{[0,i)}). \tag{1}$$

The cache holds the conditioning context $\boldsymbol{x}_{[0,i)}$, and how it is built during training defines the recipe. Teacher Forcing (Gao et al., 2024; Hu et al., 2024; Jin et al., 2024; Zhang et al., 2025b) and Diffusion Forcing (Chen et al., 2024; Yin et al., 2025; Teng et al., 2025; Song et al., 2025) condition on clean or noised ground-truth (GT) frames. At test time no GT is available, so the cache instead holds the model's own generations, opening a train-test gap. Self Forcing (Huang et al., 2025a; Cui et al., 2025; Yin et al., 2024a) and its successors (Zhu et al., 2026; Yang et al., 2025; Liu et al., 2025) train on the model's own generations and close this gap, but only within a finite training horizon.

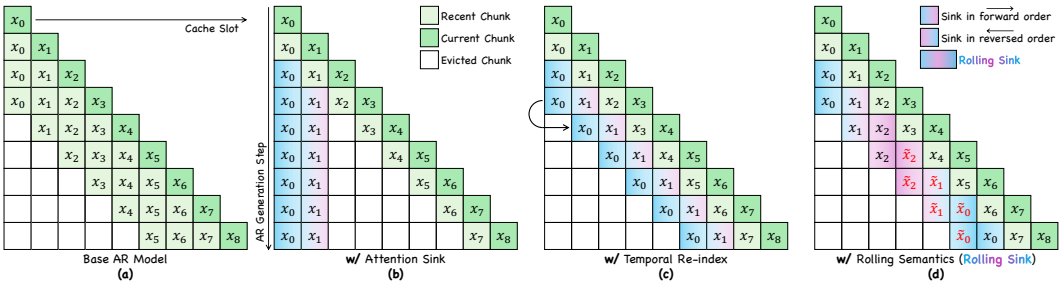

Figure 2: **Cache analysis and derivation of Rolling Sink. (a)** The base AR model uses a bounded recent cache. Tested beyond the training horizon, its temporal indices still slide and its semantics still evolve with the rollout, but the cached content gradually drifts. **(b)** Attention Sink reuses low-drift chunks generated within the training horizon, restoring low-drift content but leaving the sink chunks fixed in temporal indices and semantics. **(c)** Temporal Re-index assigns these sink chunks to the recent window just before the current chunk, restoring sliding temporal indices. **(d)** Rolling Semantics traverses the low-drift chunks in a forward-reverse schedule, then reads a sliding semantic window from this traversal to restore evolving semantics. One recent chunk is retained to keep local motion smooth. Under a bounded budget, Rolling Sink keeps the within-horizon cache behavior during long rollouts, narrowing the gap between limited-horizon training and open-ended testing.

Table 1: **Cache mechanism comparison of long-horizon AR methods.** We view the horizon gap as a domain shift on the AR cache. The base is trained only on within-horizon caches, so keeping the test cache in-distribution means matching the three factors the model conditions on: content, temporal position, and semantic trajectory. The first row is the frozen base with no test-time policy. The other rows are training-free policies applied to it under the same bounded cache. Cells restate each method's reported mechanism (*drifted*: off-distribution content, *pinned*: static sink, *rolling*: evolving sink). Attention sink (low-drift content) and RoPE re-indexing (temporal position) come from streaming language models (Xiao et al., 2023) and prior training-free horizon-extension work (Yi et al., 2025; Yesiltepe et al., 2025). But only Rolling Sink matches all three factors, and Rolling Semantics (*rolling*) is its one new component.

Method	Training-free	Cache factors beyond the training horizon		
		Content	Temporal position	Semantic trajectory
Self / Causal Forcing (Huang et al., 2025a; Zhu et al., 2026)	✗	drifted	sliding	drifted
∞-RoPE / FLEX (Yesiltepe et al., 2025; Li et al., 2026)	✓	drifted	extrapolated	drifted
Deep Forcing (Yi et al., 2025)	✓	low-drift	re-indexed	pinned
MemRoPE (Kim et al., 2026)	✓	low-drift	re-indexed	pinned
Rolling Sink (ours)	✓	low-drift	re-indexed	rolling

3.2 Key Issue: Keeping the Test Cache In-Distribution

The AR conditional $p_\theta(\boldsymbol{x}_i \mid \mathcal{C})$ is trained only on caches $\mathcal{C}$ from within-horizon rollouts. Beyond the horizon the cache is out-of-distribution, so the degradation is the ordinary cost of running a model off its training distribution, with exposure bias (Bengio et al., 2015; Lamb et al., 2016) as one instance. We cannot train on unbounded lengths, so the only test-time lever is to shrink this gap: make the test cache resemble the caches the model saw in training.

The cache enters the model through attention over its chunks. Each chunk thus contributes three things the model reads: content (the attended latent values), temporal position (the RoPE index), and, across chunks, a semantic trajectory. Matching the training distribution therefore reduces to matching these three factors: ① **low-drift content**, chunks that look like valid within-horizon frames; ② **sliding temporal indices**, the same recent window of positions as in training; and ③ **evolving semantics**, a trajectory that moves forward as in a within-horizon rollout. These are the factors the conditional depends on, not hand-picked properties. Beyond its horizon a frozen base keeps the last two but drifts on the first, which pushes the whole cache off-distribution.

We restore the three factors in order under the bounded budget of K chunks, content first: matching positions or trajectory over corrupted chunks does not help (Sec. 3.3). Fig. 2 shows the four stages, and Tab. 1 places them against prior methods.

3.3 ROLLING SINK

We recover the three factors of Sec. 3.2 in order under the bounded budget. Attention Sink (Sec. 3.3.1) reuses the attention-sink mechanism (Xiao et al., 2023) to restore low-drift content but leaves the sink static. Temporal Re-index (Sec. 3.3.2) applies the RoPE realignment of prior work (Yi et al., 2025; Yesiltepe et al., 2025) to restore sliding temporal indices. Rolling Semantics (Sec. 3.3.3), new to this paper, restores evolving semantics. The first two adapt existing mechanisms. The contribution is combining all three into one bounded-cache policy plus the Rolling Semantics schedule.

3.3.1 ATTENTION SINK: LOW-DRIFT CONTENT

We start from a base AR model, where each AR step generates a chunk $\boldsymbol{x}_i$ of n latent frames, $\boldsymbol{x}_i[k] = \boldsymbol{z}_{in+k}$ for $k \in [0, n)$[2], where $\boldsymbol{z}$ denotes a latent frame[3]. Within the training horizon the context is the full prefix $\boldsymbol{x}_{[0,i)}$. Beyond it, the base keeps only a bounded recent cache of size K:

$$p_\theta\left(\boldsymbol{x}_i \mid \boldsymbol{x}_{[i-K,i)}\right),\ i \in [K, \infty). \tag{2}$$

Its temporal indices slide and its semantics evolve with the rollout, but its chunks gradually drift, degrading quality once out-of-distribution (i.e., beyond the training horizon).

Chunks made within the training horizon are usually low-drift, since the model was optimized in that regime. To restore low-drift cached content, we reuse the attention-sink mechanism (Xiao et al., 2023) and reserve the first S cache slots for a static set of early within-horizon chunks, which we call *sink chunks*:

$$p_\theta\left(\boldsymbol{x}_i \mid \texttt{Cat}\left(\boldsymbol{x}_{[0,S)}, \boldsymbol{x}_{[i-(K-S),i)}\right)\right),\ i \in [K, \infty), \tag{3}$$

where $S \in [0, K]$ is the sink size and $\texttt{Cat}(\cdot)$ concatenates the sink and recent chunks. This restores low-drift content, but leaves the sink chunks static in both temporal indices and semantics.

3.3.2 TEMPORAL RE-INDEX: SLIDING TEMPORAL INDICES

The sink chunks keep their original temporal indices throughout the rollout, unlike the within-horizon cache, where chunks occupy a sliding window just before the current chunk. We re-index the sink chunks to this sliding window while keeping their low-drift content unchanged.

Specifically, we use $\boldsymbol{x}_i^j$ to denote chunk $\boldsymbol{x}_i$ embedded with temporal index j:

$$\begin{aligned} \boldsymbol{x}_i^j &= \texttt{RoPE}(\boldsymbol{x}_i, j), \\ \boldsymbol{x}_i^j[k] &= \texttt{RoPE}(\boldsymbol{z}_{in+k}, jn+k),\ k \in [0, n). \end{aligned} \tag{4}$$

When no superscript is specified, we have $j \equiv i$. Following Eq. 3, Temporal Re-index assigns the S sink chunks to the sliding temporal window $[i-K, i-(K-S))$:

$$\begin{aligned} &p_\theta\left(\boldsymbol{x}_i \mid \texttt{Cat}\left(\boldsymbol{x}_{[0,S)}^{[i-K,i-(K-S))}, \boldsymbol{x}_{[i-(K-S),i)}\right)\right), \\ &\text{where } i \in [K, \infty),\ \boldsymbol{x}_{[0,S)}^{[i-K,i-(K-S))} = \left\{\boldsymbol{x}_l^{i-K+l}\right\}_{l \in [0,S)}. \end{aligned} \tag{5}$$

Temporal Re-index restores sliding temporal indices while preserving low-drift content, but the sink's semantics are still fixed to the same early chunks $\boldsymbol{x}_{[0,S)}$.

3.3.3 ROLLING SEMANTICS: EVOLVING SEMANTIC CONTENT

Temporal Re-index still pins the sink to the same early chunks $\boldsymbol{x}_{[0,S)}$, so its semantics stay fixed while the rollout moves on. Rolling Semantics lets that content evolve. It treats the finite bank of K low-drift chunks $\boldsymbol{x}_{[0,K)}$, generated within the training horizon, as ordered samples along a semantic axis. At each step it reads a sliding window of S consecutive positions from this axis. The

[2]Throughout this paper, all interval values are integers unless otherwise specified, *e.g.*, $[0, n)$ indicates $\{0, 1, \cdots, n-1\}$.

[3]Each latent frame corresponds to multiple RGB frames (Wan et al., 2025).

Table 2: **1-minute AR video synthesis: comparison with training-free methods.** We compare Rolling Sink with training-free horizon-extension methods on two 5s-trained bases, Self Forcing (Huang et al., 2025a) and Causal Forcing (Zhu et al., 2026), using `VBench-Long` (Huang et al., 2024; 2025b; Zheng et al., 2025). Rows are grouped by base model, each group holds the same base and inference setting. Rolling Sink attains the best averaged rank in both groups. The best and second-best scores within each group are in **bold** and underline. *Forward-only* drops the reverse leg of Eq. 7, cycling the bank forward only ($\texttt{Roll}(\boldsymbol{x}_{[0,K)})[l] = \boldsymbol{x}^{l}_{l \bmod K}$), so the sink repeats with period K.

Method	VBench-Long Dimensions ↑																Avg
	SubCon	BgCon	AesQl	ImgQl	ObjCls	MulObj	Col	SpaRel	Sce	TmpSty	OvrCon	HumAct	TmpFli	MoSmo	DyDg	AppSty	Rank↓
Base AR model: Self Forcing																	
Self Forcing	96.79	96.53	59.16	69.80	86.80	36.39	64.33	71.21	10.79	22.20	19.91	68.86	97.63	98.14	48.57	**20.99**	5.06
Deep Forcing	97.83	95.92	60.20	69.48	**100.00**	58.34	80.35	99.07	18.10	24.34	22.74	**88.57**	97.57	98.33	75.00	19.33	3.75
∞-RoPE	98.17	96.23	61.70	72.09	**100.00**	58.20	80.93	98.11	17.46	23.41	20.97	83.71	97.62	97.81	**76.07**	20.21	3.62
MemRoPE	98.41	96.20	58.49	**72.21**	99.69	**81.19**	**92.06**	98.56	**25.16**	23.93	21.82	75.88	97.94	98.56	62.87	19.50	3.25
Rolling Sink	**98.58**	**96.94**	**63.08**	69.68	**100.00**	69.98	80.23	**100.00**	21.59	**25.03**	23.16	78.00	**98.16**	**98.65**	74.69	18.91	**2.19**
(Forward Only)	98.45	96.75	62.16	68.95	**100.00**	68.64	81.73	99.99	19.24	24.95	**23.35**	75.33	98.11	98.48	75.36	18.70	2.75
Base AR model: Causal Forcing																	
Causal Forcing	95.65	96.11	52.73	62.86	69.02	32.23	74.53	65.91	13.02	18.39	17.73	50.57	**96.30**	97.17	54.64	**21.13**	4.25
Deep Forcing	97.49	95.18	61.05	68.99	89.50	35.91	84.33	92.09	16.83	22.19	20.39	75.14	95.38	97.48	93.57	20.34	3.25
MemRoPE	97.00	94.80	60.83	**71.26**	94.89	40.92	79.13	96.98	16.01	**23.24**	21.11	76.47	93.53	97.39	88.24	20.55	3.00
∞-RoPE	96.99	95.40	**62.68**	70.16	96.12	39.88	80.14	95.77	21.90	23.16	21.61	82.29	95.44	95.46	**98.93**	20.02	2.69
Rolling Sink	**98.16**	**96.38**	62.42	70.45	**98.66**	**74.96**	**89.02**	**98.52**	**22.86**	21.86	**23.30**	**84.29**	96.24	**98.04**	78.57	20.00	**1.81**

Table 3: **5-minute AR video synthesis: comparison with training-free methods.** This setting is harder than Tab. 2, since drift has longer to accumulate. Rolling Sink again attains the best averaged rank on both the Self Forcing (Huang et al., 2025a) and Causal Forcing (Zhu et al., 2026) bases. Following Tab. 2, rows are grouped by base model. Best and second-best within each group are in **bold** and underline.

Method	VBench-Long Dimensions ↑																Avg
	SubCon	BgCon	AesQl	ImgQl	ObjCls	MulObj	Col	SpaRel	Sce	TmpSty	OvrCon	HumAct	TmpFli	MoSmo	DyDg	AppSty	Rank↓
Base AR model: Self Forcing																	
Self Forcing	94.24	96.10	42.89	57.01	33.39	14.27	61.05	35.70	4.30	11.32	11.39	27.10	98.20	**98.65**	24.19	**20.53**	4.88
MemRoPE	96.69	94.94	57.94	71.76	99.64	62.66	73.61	95.46	23.66	21.35	21.82	80.65	97.79	98.47	65.32	19.47	4.25
∞-RoPE	97.14	94.43	59.00	**71.82**	89.15	49.09	82.41	95.21	27.24	21.97	19.99	**90.32**	97.88	97.76	70.56	19.72	3.50
Deep Forcing	96.76	94.59	60.54	69.29	99.90	60.52	**86.65**	98.97	**33.33**	23.72	22.16	83.55	97.88	97.67	**78.23**	19.57	3.25
Rolling Sink	**98.04**	**96.29**	**62.96**	69.87	**100.00**	**72.84**	78.83	**100.00**	26.16	**25.33**	23.10	87.10	**98.32**	98.59	64.11	18.91	**2.12**
(Forward Only)	97.82	96.03	62.47	69.23	**100.00**	71.57	81.13	**100.00**	25.47	24.98	**23.30**	86.57	98.16	98.53	65.03	18.75	2.81
Base AR model: Causal Forcing																	
Causal Forcing	92.35	94.94	40.30	55.12	29.11	8.33	75.05	21.40	4.66	11.33	11.10	38.39	**96.97**	96.70	31.45	**20.80**	4.19
Deep Forcing	94.74	92.81	58.37	68.14	85.54	29.84	79.13	85.93	16.13	21.84	20.31	**86.13**	94.90	96.95	89.52	20.69	3.25
MemRoPE	94.16	92.49	59.61	68.85	91.83	38.10	71.88	88.15	15.05	**25.08**	22.23	80.00	93.50	96.75	93.95	20.46	3.12
∞-RoPE	95.19	93.32	**62.86**	70.15	**95.02**	37.36	81.51	92.03	**26.16**	24.98	23.21	80.00	95.33	95.41	**99.60**	20.39	2.25
Rolling Sink	**97.09**	**95.57**	61.97	**70.32**	91.79	**72.80**	**88.29**	**97.84**	17.56	21.55	**23.27**	78.39	96.26	**98.16**	84.68	20.12	**2.12**

sink content then shifts by one position per chunk instead of freezing, mirroring the within-horizon rollout where the semantic window moves chunk by chunk. The conditional distribution becomes:

$$p_\theta\left(\boldsymbol{x}_i \mid \texttt{Cat}\left(\texttt{Roll}(\boldsymbol{x}_{[0,K)})_{[i-K,i-(K-S))}, \boldsymbol{x}_{[i-(K-S),i)}\right)\right), \quad \text{where } i \in [K,\infty),\ \texttt{Roll}(\boldsymbol{x}_{[0,K)})_{[a,b)} = \left\{\texttt{Roll}(\boldsymbol{x}_{[0,K)})[l]\right\}_{l\in[a,b)}. \tag{6}$$

The bank is finite, so a long rollout eventually revisits its chunks: naive cycling does this by jumping from the last chunk $\boldsymbol{x}_{K-1}$ back to $\boldsymbol{x}_0$ every K steps. Rolling Semantics instead traverses the bank forward then backward, reversing the frame order within a chunk on the backward pass, so successive read-outs stay adjacent and no periodic jump appears:

$$\texttt{Roll}(\boldsymbol{x}_{[0,K)})[l] = \begin{cases} \boldsymbol{x}^{l}_{l \bmod K} & \text{if } \left\lfloor \dfrac{l}{K} \right\rfloor \bmod 2 = 0, \\ \tilde{\boldsymbol{x}}^{l}_{(K-1)-(l \bmod K)} & \text{if } \left\lfloor \dfrac{l}{K} \right\rfloor \bmod 2 = 1, \end{cases} \tag{7}$$

where $l \in [0,\infty)$ and $\tilde{\boldsymbol{x}}^{j}_{i}$ is the frame-reversed form of $\boldsymbol{x}^{j}_{i}$, with $\tilde{\boldsymbol{x}}^{j}_{i}[k] = \texttt{RoPE}(\boldsymbol{z}_{in+(n-1-k)}, jn+k)$ for $k \in [0,n)$.

Retaining one recent chunk beside the rolling sink anchors each prediction to the most recent context, keeping local motion smooth and the output moving forward rather than replaying the bank (App. D and E). The backward pass reorders frames only inside the sink chunks, which supply content and semantics rather than the visible motion. The motion is carried by the recent chunk, which is not reversed, so the reversal does not play the visible dynamics backward. Rolling Sink is the resulting policy: under the bounded budget it holds low-drift content, sliding temporal indices,

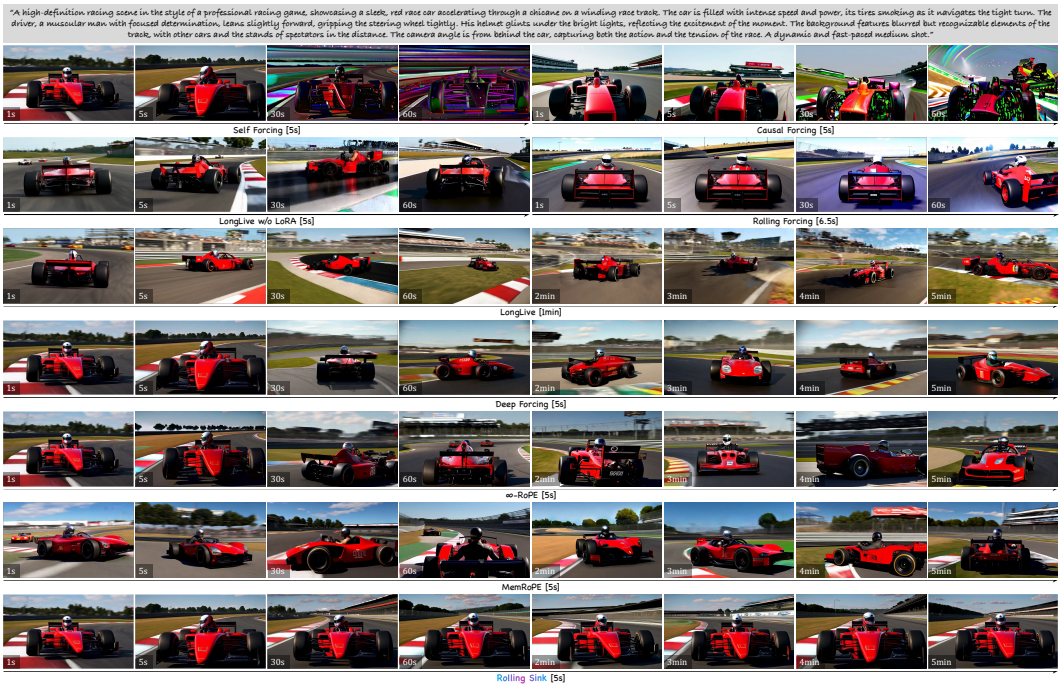

Figure 3: **Qualitative comparison on long-horizon AR video synthesis.** We compare Rolling Sink with training-based and training-free horizon-extension methods; each method's training duration is in square brackets. For Rolling Sink and the training-free baselines, the base model is Self Forcing (Huang et al., 2025a). Rolled out beyond their training horizons, the baselines accumulate drift: oversaturated colors, distorted structure, subject change, and fading motion. Rolling Sink holds subject, structure, and motion more stable across the rollout while using only a 5s base, including against baselines trained on minute-long clips.

and evolving semantics together. This re-scheduling of within-horizon content narrows the cache-behavior mismatch between limited-horizon training and open-ended testing, but cannot fully close it (Sec. 5). App. F gives the full pseudo-code.

4 Experiments

4.1 Experimental Setup

Implementation. Rolling Sink is training-free. We apply it to two 5s-trained bases, Self Forcing (Huang et al., 2025a) and Causal Forcing (Zhu et al., 2026), and reuse each base's sampling configuration: the same few-step denoising and the same bounded cache budget $K = 6$. We set $S/K = 5/6$ (ablated in Sec. 4.4).

Protocol. We evaluate two rollout lengths, 1 minute and 5 minutes, with `VBench-Long` (Huang et al., 2024; 2025b; Zheng et al., 2025), which scores long videos on 16 dimensions. We set its `clip_length` to 2.0 for the 1-minute setting and 10.0 for the 5-minute one. We report every dimension and the averaged rank, where a lower rank is better. Averaged rank is computed within each same-base group. We score a fixed subset of 10 prompts per dimension, drawn once from the original 70 to 100 and reused across all runs. App. G gives the full dimension names and this sampling protocol. The training-free comparison holds the same base model and inference setting. Qualitatively, we compare rollouts at matched timestamps for color shift, identity change, structural collapse, and motion loss.

Baselines. We compare against two groups. The training-free horizon-extension group includes Deep Forcing (Yi et al., 2025), ∞-RoPE (Yesiltepe et al., 2025), and MemRoPE (Kim et al., 2026). The training-based group includes Self Forcing (Huang et al., 2025a), Causal Forcing (Zhu et al., 2026), Rolling Forcing (Liu et al., 2025), and LongLive (Yang et al., 2025).

Table 4: **Comparison with training-based AR baselines.** We compare Rolling Sink with training-based AR baselines at 1-minute and 5-minute rollouts on VBench-Long (Huang et al., 2024; 2025b; Zheng et al., 2025). Superscripts give each method's training duration; the Rolling Sink subscript gives its base, Self Forcing (SF) (Huang et al., 2025a) or Causal Forcing (CF) (Zhu et al., 2026). Rolling Sink on the SF base reaches the best averaged rank at both lengths, even ahead of LongLive (Yang et al., 2025) trained on 1-minute clips, while using no extra training on a 5s base. The CF variant ranks third, behind LongLive-1min. The best and second-best scores are in **bold** and underline.

Method	VBench-Long Dimensions ↑															Avg
	SubCon	BgCon	AesQl	ImgQl	ObjCls	MulObj	Col	SpaRel	Sce	TmpSty	OvrCon	HumAct	TmpFli	MoSmo	DyDg	AppSty	Rank↓
1-minute AR Video Synthesis																	
Causal Forcing^{5s}	95.65	96.11	52.73	62.86	69.02	32.23	74.53	65.91	13.02	18.39	17.73	50.57	96.30	97.17	54.64	**21.13**	6.25
Self Forcing^{5s}	96.79	96.53	59.16	69.80	86.80	36.39	64.33	71.21	10.79	22.20	19.91	68.86	97.63	98.14	48.57	20.99	4.81
LongLive$^{5s}_{\text{w/o LoRA}}$	96.68	95.88	58.50	65.19	97.80	58.02	77.12	96.83	**25.40**	23.98	21.60	**88.57**	96.43	97.30	70.71	20.18	4.50
Rolling Forcing$^{6.5s}$	97.95	96.22	60.68	68.09	**100.00**	62.25	84.97	99.58	24.13	24.12	20.49	74.00	97.33	97.87	32.86	20.23	3.88
Rolling Sink$^{5s}_{\text{CF}}$	98.16	96.38	62.42	**70.45**	98.66	74.96	89.02	98.52	22.86	21.86	**23.30**	84.29	96.24	98.04	**78.57**	20.00	3.25
LongLive^{1min}	98.40	96.50	62.56	69.47	**100.00**	**88.64**	**98.66**	**100.00**	15.87	23.29	23.21	80.57	**98.18**	98.48	55.00	18.77	2.69
Rolling Sink$^{5s}_{\text{SF}}$	**98.58**	**96.94**	**63.08**	69.68	**100.00**	69.98	80.23	**100.00**	21.59	**25.03**	23.16	78.00	98.16	**98.65**	74.69	18.91	**2.38**
5-minute AR Video Synthesis																	
Causal Forcing^{5s}	92.35	94.94	40.30	55.12	29.11	8.33	75.05	21.40	4.66	11.33	11.10	38.39	96.97	96.70	31.45	20.80	6.06
Self Forcing^{5s}	94.24	96.10	42.89	57.01	33.39	14.27	61.05	35.70	4.30	11.32	11.39	27.10	98.20	**98.65**	24.19	20.53	5.31
LongLive$^{5s}_{\text{w/o LoRA}}$	93.93	94.27	57.18	64.31	96.65	69.98	73.02	96.97	20.79	24.35	21.50	**95.48**	96.87	97.66	73.79	**20.86**	4.31
Rolling Forcing$^{6.5s}$	96.72	95.09	58.09	67.70	**100.00**	71.73	77.63	98.57	**34.41**	23.84	20.20	71.94	97.58	98.04	31.05	20.21	3.94
Rolling Sink$^{5s}_{\text{CF}}$	97.09	95.57	61.97	**70.32**	91.79	72.80	**88.29**	97.84	17.56	21.55	**23.27**	78.39	96.26	98.16	**84.68**	20.12	3.44
LongLive^{1min}	96.91	96.01	**63.70**	69.78	**100.00**	**76.90**	82.80	**100.00**	20.07	24.57	22.66	86.13	**98.35**	98.46	59.68	18.54	2.63
Rolling Sink$^{5s}_{\text{SF}}$	**98.04**	**96.29**	62.96	69.87	**100.00**	72.84	78.83	**100.00**	26.16	**25.33**	23.10	87.10	98.32	98.59	64.11	18.91	**2.06**

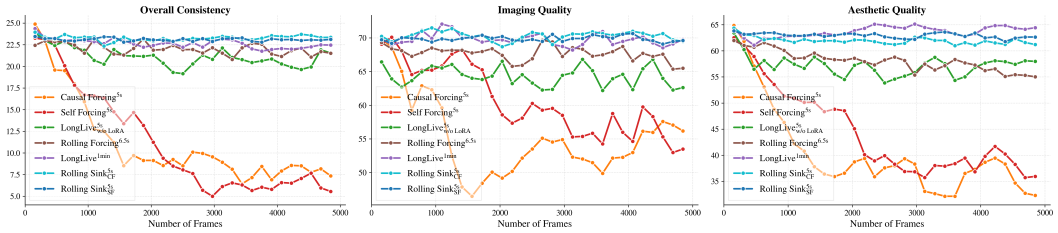

Figure 4: **Drift over the 5-minute rollout.** We track three VBench-Long (Huang et al., 2024; 2025b; Zheng et al., 2025) dimensions, Overall Consistency, Imaging Quality, and Aesthetic Quality, along the generation trajectory. Baselines degrade or fluctuate as frames accumulate, reflecting drift. Rolling Sink stays higher and flatter across the rollout, keeping quality high while slowing drift.

4.2 Comparison with Training-Free Methods

Tab. 2 and Tab. 3 compare Rolling Sink with training-free horizon-extension methods. Rolling Sink attains the best averaged rank in all four groups (2.19 and 1.81 at 1 minute on Self Forcing and Causal Forcing, 2.12 and 2.12 at 5 minutes), consistent across both lengths and bases. This supports treating the horizon gap as cache-behavior matching rather than a single-axis fix. The Forward-only rows isolate the schedule: forward-only cycling of the bank trails the full forward-reverse schedule on averaged rank, so the reverse leg helps beyond simple cycling. Averaged rank aggregates 16 dimensions, so the lead is an aggregate one under the evaluated protocol, not a win on every dimension.

Per dimension, the gains concentrate on the consistency, quality, and temporal-stability axes: on the 1-minute Self Forcing base, Rolling Sink leads on subject and background consistency, aesthetic quality, spatial relationship, temporal style, temporal flickering, and motion smoothness. It trades off elsewhere, with MemRoPE leading on imaging quality, multiple objects, color, and scene, Deep Forcing on human action, ∞-RoPE on dynamic degree, and the frozen base on appearance style by about two points, so the averaged rank is a balance rather than a uniform lead.

4.3 Comparison with Training-Based Baselines

Tab. 4 adds training-based baselines, several trained on longer clips (6.5s for Rolling Forcing, 1 minute for LongLive). Rolling Sink on the 5s Self Forcing base reaches the best averaged rank at both lengths (2.38 and 2.06), ahead of LongLive trained on 1-minute clips (2.69 and 2.63), despite using no extra training and a shorter horizon.

4.4 Cache-Behavior Ablation

Fig. 5 scores each stage of the derivation (Sec. 3.3) on `VBench-Long` (Huang et al., 2024; Zheng et al., 2025; Huang et al., 2025b), plotting *Video Quality*, the mean over all 16 dimensions (the Avg Score of Tab. S2 and S3). Adding each cache behavior raises this score at both 1 and 5 minutes. Attention Sink provides the largest single gain by restoring low-drift content. Temporal Re-index and Rolling Semantics add smaller further gains, and Rolling Semantics reaches the best final score at the default $S/K = 5/6$. The full per-dimension numbers under every sink ratio are in Tab. S2 and S3. We use $S/K = 5/6$ as the default, its best-performing setting[4].

4.5 Qualitative Comparison and Drift Analysis

Fig. 3 shows rollouts against both baseline groups (base model Self Forcing). ∞-RoPE extends positions but does not restore low-drift content, and Deep Forcing and MemRoPE retain a sink or memory prefix with fixed semantics, so all three still show color shift, structural change, or subject drift. Training-based baselines also drift once rolled far past their trained length. Rolling Sink holds subject, structure, and motion more stable across the rollout while using only a 5s base.

Fig. 4 tracks three `VBench-Long` dimensions along the 5-minute trajectory: baselines degrade or fluctuate as frames accumulate, whereas Rolling Sink stays higher and flatter. Keeping the cache near its within-horizon behavior both raises quality and slows drift.

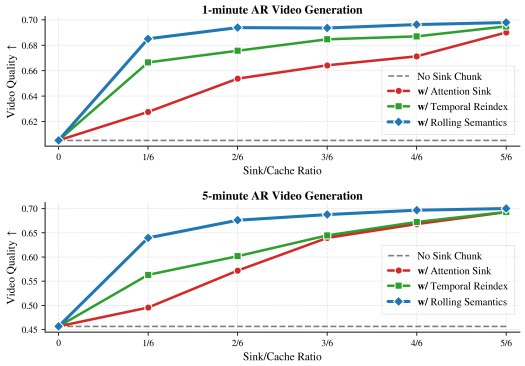

Figure 5: **Cache-behavior ablation.** *Video Quality* (mean over all `VBench-Long` dimensions) rises as each cache behavior is added, best at $S/K = 5/6$.

4.6 Inference Efficiency

Rolling Sink keeps the base's cache size and denoising steps, so its throughput tracks the base: 5.35 vs. 5.38 FPS on an RTX 4090 at the 1-minute setting, a 0.56% change (App. B). The bounded cache also holds GPU memory constant, since K does not grow with rollout length.

5 Conclusion and Limitations

We cast long-horizon AR drift as a domain shift on the cache and propose Rolling Sink, a training-free bounded-cache policy that matches the within-horizon cache along content, position, and semantics. It reuses attention sink and temporal re-indexing and adds the Rolling Semantics schedule. It attains the best averaged `VBench-Long` rank at 1- and 5-minute rollouts on two 5s bases at about 0.56% throughput change, and sustains a 30-minute qualitative rollout.

Limitations. Under limited-horizon training, the gap can only be reduced, not closed: the base never learned the long-range content that open-ended testing needs. Rolling Sink matches content and position near-exactly (reusing low-drift chunks, re-indexed to the exact window) but only approximates semantics: the finite bank cannot synthesize new long-range content, it reads out the same chunks periodically, and its reverse legs run the sink's semantics backward (a partial time-reversal). This residual needs more training or data, not a better cache. Matching all three factors (Tab. 1) is the closest a training-free policy can come. We analyze return-to-start repetition in App. D and periodic replay and progression in App. E. Several directions remain open, such as behavior on individual irreversible events (an object falling, liquid pouring, something breaking).

[4] $S/K = 5/6$ keeps five rolling sink chunks and one recent chunk under the shared budget $K = 6$. We sweep S/K within this budget, since other budgets would require different bases.

6 AI USE STATEMENT

We used generative AI tools solely for polishing grammar and sentence structures, to improve readability, clarity, and fluency. They made no contribution to the original research content. All the scientific and technical content of this paper was written entirely by humans. We have reviewed all AI-assisted work and we take responsibility for all the final content of this work.

REFERENCES

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28, 2015.

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. *arXiv preprint arXiv:2311.15127*, 2023.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, et al. Skyreels-v2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025.

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.

Haoge Deng, Ting Pan, Haiwen Diao, Zhengxiong Luo, Yufeng Cui, Huchuan Lu, Shiguang Shan, Yonggang Qi, and Xinlong Wang. Autoregressive video generation without vector quantization. *arXiv preprint arXiv:2412.14169*, 2024.

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. *arXiv preprint arXiv:2411.16375*, 2024.

Google. Introducing veo 3, our video generation model with expanded creative controls – including native audio and extended videos. `https://deepmind.google/models/veo/`, 2025.

Yuwei Guo, Ceyuan Yang, Ziyan Yang, Zhibei Ma, Zhijie Lin, Zhenheng Yang, Dahua Lin, and Lu Jiang. Long context tuning for video generation. *arXiv preprint arXiv:2503.10589*, 2025.

Jing He, Haodong Li, Yongzhe Hu, Guibao Shen, Yingjie Cai, Weichao Qiu, and Ying-Cong Chen. Disenvisioner: Disentangled and enriched visual prompt for customized image generation. *arXiv preprint arXiv:2410.02067*, 2024a.

Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Zhang, Bingbing Liu, and Ying-Cong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. *arXiv preprint arXiv:2409.18124*, 2024b.

Jing He, Haodong Li, Mingzhi Sheng, and Ying-Cong Chen. Lotus-2: Advancing geometric dense prediction with powerful image generative model. *arXiv preprint arXiv:2512.01030*, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jinyi Hu, Shengding Hu, Yuxuan Song, Yufei Huang, Mingxuan Wang, Hao Zhou, Zhiyuan Liu, Wei-Ying Ma, and Maosong Sun. Acdit: Interpolating autoregressive conditional modeling and diffusion transformer. *arXiv preprint arXiv:2412.07720*, 2024.

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

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

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

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. *arXiv preprint arXiv:2410.05954*, 2024.

Youngrae Kim, Qixin Hu, C-C Jay Kuo, and Peter A Beerel. Memrope: Training-free infinite video generation via evolving memory tokens. *arXiv preprint arXiv:2603.12513*, 2026.

Kling. Kling video 2.6 – kling's first "native audio" model official launched! `https://app.klingai.com/global/release-notes/c605hp1tzd`, 2025.

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. *arXiv preprint arXiv:2412.03603*, 2024.

Stanley Kubrick. The shining. `https://en.wikipedia.org/wiki/The_Shining_(film)`, 1980.

Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. *Advances in neural information processing systems*, 29, 2016.

Jia Li, Xiaomeng Fu, Xurui Peng, Weifeng Chen, Youwei Zheng, Tianyu Zhao, Jiexi Wang, Fangmin Chen, Xing Wang, and Hayden Kwok-Hay So. Train short, inference long: Training-free horizon extension for autoregressive video generation. *arXiv preprint arXiv:2602.14027*, 2026.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. *arXiv preprint arXiv:2509.25161*, 2025.

Xiaofeng Mao, Shaohao Rui, Kaining Ying, Bo Zheng, Chuanhao Li, Mingmin Chi, and Kaipeng Zhang. Packforcing: Short video training suffices for long video sampling and long context inference. *arXiv preprint arXiv:2603.25730*, 2026.

Steve McQueen. Hunger. `https://en.wikipedia.org/wiki/Hunger_(2008_film)`, 2008.

OpenAI. Sora 2 is here. `https://openai.com/index/sora-2/`, 2025.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

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

Shuhuai Ren, Shuming Ma, Xu Sun, and Furu Wei. Next block prediction: Video generation via semi-autoregressive modeling. *arXiv preprint arXiv:2502.07737*, 2025.

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*, pp. 10684–10695, 2022.

Runway. Introducing runway gen-4.5: A new frontier for video generation. https://runwayml.com/research/introducing-runway-gen-4.5, 2025.

Team Seedance, De Chen, Liyang Chen, Xin Chen, Ying Chen, Zhuo Chen, Zhuowei Chen, Feng Cheng, Tianheng Cheng, Yufeng Cheng, et al. Seedance 2.0: Advancing video generation for world complexity. *arXiv preprint arXiv:2604.14148*, 2026.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. *arXiv preprint arXiv:2502.06764*, 2025.

Hansi Teng, Hongyu Jia, Lei Sun, Lingzhi Li, Maolin Li, Mingqiu Tang, Shuai Han, Tianning Zhang, WQ Zhang, Weifeng Luo, et al. Magi-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211*, 2025.

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

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, Jiayu Wang, Jingfeng Zhang, Jingren Zhou, Jinkai Wang, Jixuan Chen, Kai Zhu, Kang Zhao, Keyu Yan, Lianghua Huang, Mengyang Feng, Ningyi Zhang, Pandeng Li, Pingyu Wu, Ruihang Chu, Ruili Feng, Shiwei Zhang, Siyang Sun, Tao Fang, Tianxing Wang, Tianyi Gui, Tingyu Weng, Tong Shen, Wei Lin, Wei Wang, Wei Wang, Wenmeng Zhou, Wente Wang, Wenting Shen, Wenyuan Yu, Xianzhong Shi, Xiaoming Huang, Xin Xu, Yan Kou, Yangyu Lv, Yifei Li, Yijing Liu, Yiming Wang, Yingya Zhang, Yitong Huang, Yong Li, You Wu, Yu Liu, Yulin Pan, Yun Zheng, Yuntao Hong, Yupeng Shi, Yutong Feng, Zeyinzi Jiang, Zhen Han, Zhi-Fan Wu, and Ziyu Liu. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Mingqiang Wu, Weilun Feng, Zhefeng Zhang, Haotong Qin, Yuqi Li, Guoxin Fan, Xiaokun Liu, Zhulin An, Libo Huang, Yongjun Xu, et al. Echo-forcing: A scene memory framework for interactive long video generation. *arXiv preprint arXiv:2605.16003*, 2026.

Xunzhi Xiang, Zixuan Duan, Guiyu Zhang, Haiyu Zhang, Zhe Gao, Junta Wu, Shaofeng Zhang, Tengfei Wang, Qi Fan, and Chunchao Guo. Pathwise test-time correction for autoregressive long video generation. *arXiv preprint arXiv:2602.05871*, 2026.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, et al. Longlive: Real-time interactive long video generation. *arXiv preprint arXiv:2509.22622*, 2025.

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. *arXiv preprint arXiv:2408.06072*, 2024.

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. *arXiv preprint arXiv:2511.20649*, 2025.

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

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo 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*, pp. 6613–6623, 2024b.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22963–22974, 2025.

Hongfei Zhang, Harold Haodong Chen, Chenfei Liao, Jing He, Zixin Zhang, Haodong Li, Yihao Liang, Kanghao Chen, Bin Ren, Xu Zheng, et al. Dvd: Deterministic video depth estimation with generative priors. In *ICML 2026 Workshop on Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning*, 2026.

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William T Freeman, and Hao Tan. Test-time training done right. *arXiv preprint arXiv:2505.23884*, 2025b.

Dian Zheng, Ziqi Huang, Hongbo Liu, Kai Zou, Yinan He, Fan Zhang, Yuanhan Zhang, Jingwen He, Wei-Shi Zheng, Yu Qiao, and Ziwei Liu. VBench-2.0: Advancing video generation benchmark suite for intrinsic faithfulness. *arXiv preprint arXiv:2503.21755*, 2025.

Jinsong Zhou, Yihua Du, Xinli Xu, Luozhou Wang, Zijie Zhuang, Yehang Zhang, Shuaibo Li, Xiaojun Hu, Bolan Su, and Ying-cong Chen. Videomemory: Toward consistent video generation via memory integration. *arXiv preprint arXiv:2601.03655*, 2026.

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 interactive video generation. *arXiv preprint arXiv:2602.02214*, 2026.

Supplementary Roadmap. App. A shows the 30-minute rollout above. App. B details the efficiency study behind the main-paper FPS numbers. App. C reports the full per-dimension ablations under every sink ratio. App. D discusses return-to-start repetition and the boundaries of Rolling Semantics. App. E quantifies periodic replay and content progression at the bank period. App. F gives the AR generation pseudo-code. App. G covers the `VBench-Long` protocol and the dimension legend.

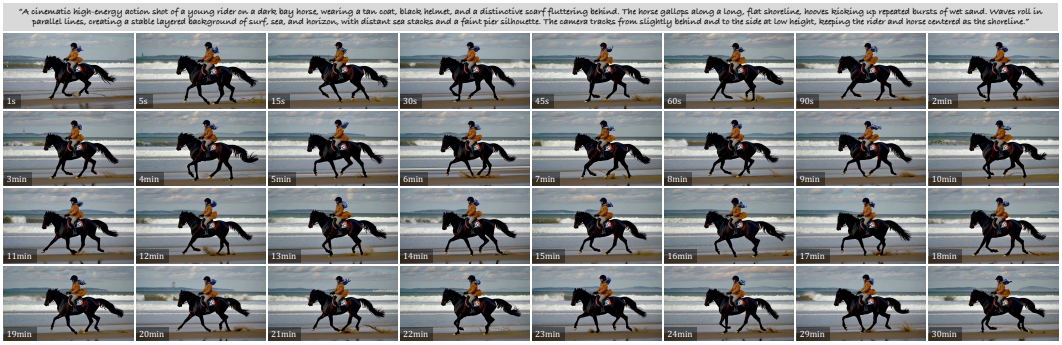

Figure S1: **30-minute AR rollout with Rolling Sink (qualitative).** Without extra training, Rolling Sink keeps a 5s-trained AR video diffusion model rolling out for 30 minutes (360× its training horizon) under a bounded cache, here on realistic and animated content. Across the shown frames, subject identity, scene structure, colors, and motion stay coherent. This is a qualitative example: it shows the policy continues to generate at this length, and does not certify quality at arbitrary duration.

A Ultra-Long AR Video Synthesis

We test Rolling Sink at 30 minutes to see how far the bounded-cache policy runs past the 1- and 5-minute settings evaluated in the main paper. Fig. S1 shows the rollout continuing on both realistic and animated content, with stable subject identity, coherent structure and color, and smooth motion across the shown frames. We present this length as a qualitative demonstration only. Automatic long-video metrics and human studies at 30 minutes are open, and fine-grained details can still degrade over such rollouts (Sec. 5).

B Efficiency Study

This section details the FPS numbers reported in Sec. 4. Rolling Sink keeps the base model's bounded cache budget and few-step denoising, and only changes which cached chunks are selected and how they are re-indexed. We measure throughput against Self Forcing (Huang et al., 2025a) on a single NVIDIA RTX 4090 GPU, under the 1-minute setting of Tab. 2: 961-frame generation, 480×832 resolution, 4 denoising steps, and a 6-chunk cache, excluding I/O. Tab. S1 reports 5.35 FPS for Rolling Sink against 5.38 FPS for Self Forcing, a 0.56% change. Because the cache holds a fixed $K = 6$ chunks, peak GPU memory stays constant as the rollout grows, unlike full-context retention whose memory scales with the number of generated chunks.

Table S1: **Efficiency comparison.** Rolling Sink introduces negligible runtime overhead under the same cache budget and denoising setting.

Method	FPS ↑
Self Forcing	5.38
Rolling Sink	5.35

C Quantitative Analysis of Rolling Sink

This section provides the full numerical results for the cache-behavior analysis in Fig. 5 of the main paper. We evaluate three progressive design steps following Sec. 3.2: restoring low-drift content

Table S2: **Full ablation on 1-minute AR video synthesis (all `VBench-Long` dimensions).** For each sink/cache ratio, we add Attention Sink, Temporal Re-index, and Rolling Semantics in turn, recovering low-drift content, sliding temporal indices, and evolving semantics. The overall score rises as each behavior is added, matching the main-paper analysis (Fig. 5). **Bold** and underline mark the best and second-best within each ratio.

Cache Strategy	VBench-Long Dimensions ↑																Avg
	SubCon	BgCon	AesQl	ImgQl	ObjCls	MulObj	Col	SpaRel	Sce	TmpSty	OvrCon	HumAct	TmpFli	MoSmo	DyDg	AppSty	Score↑
Sink/Cache Ratio: 1/6																	
No Sink Chunk	96.79	96.53	59.16	69.80	86.80	36.39	64.33	71.21	10.79	22.20	19.91	68.86	97.63	98.14	48.57	**20.99**	60.51
w/ Attention Sink	**98.70**	96.61	61.21	70.04	92.91	58.84	68.36	91.78	15.87	21.91	21.36	72.00	**98.20**	**99.03**	16.79	20.39	62.75
w/ Temporal Reindex	97.62	96.67	60.80	**70.58**	99.88	54.57	**75.89**	97.11	17.14	22.28	22.45	68.00	97.87	97.60	67.50	20.45	66.65
w/ Rolling Semantics	97.66	**96.78**	**61.52**	68.86	**100.00**	**63.88**	75.26	**98.87**	**23.49**	**24.46**	**23.08**	**77.14**	98.18	97.64	**68.93**	20.37	**68.51**
Sink/Cache Ratio: 2/6																	
No Sink Chunk	96.79	96.53	59.16	69.80	86.80	36.39	64.33	71.21	10.79	22.20	19.91	68.86	97.63	98.14	48.57	**20.99**	60.51
w/ Attention Sink	**98.76**	**96.91**	61.87	70.12	98.50	**69.91**	79.09	95.64	13.02	22.70	21.53	**83.71**	98.01	**99.12**	17.14	19.95	65.37
w/ Temporal Reindex	98.07	96.62	60.50	**70.73**	99.89	63.36	78.20	96.73	**22.86**	22.47	22.01	69.14	97.80	98.06	64.64	19.98	67.57
w/ Rolling Semantics	98.16	96.86	**62.36**	69.01	**100.00**	69.37	**79.76**	**99.45**	21.90	**24.77**	**23.25**	78.86	**98.11**	98.31	**70.36**	19.69	**69.39**
Sink/Cache Ratio: 3/6																	
No Sink Chunk	96.79	96.53	59.16	69.80	86.80	36.39	64.33	71.21	10.79	22.20	19.91	68.86	97.63	98.14	48.57	**20.99**	60.51
w/ Attention Sink	**99.03**	**96.93**	62.09	69.31	**100.00**	**80.32**	76.71	98.31	18.10	22.94	22.70	83.14	**98.22**	**99.14**	16.43	19.36	66.42
w/ Temporal Reindex	97.83	96.49	60.94	**69.95**	**100.00**	66.93	**85.73**	96.29	**21.59**	22.10	22.22	72.29	97.55	97.62	**68.57**	19.50	68.47
w/ Rolling Semantics	98.27	96.86	**62.68**	69.62	**100.00**	69.93	80.52	**99.96**	17.78	**24.80**	**23.04**	**83.71**	98.17	98.48	66.43	19.53	**69.36**
Sink/Cache Ratio: 4/6																	
No Sink Chunk	96.79	96.53	59.16	69.80	86.80	36.39	64.33	71.21	10.79	22.20	19.91	68.86	97.63	98.14	48.57	**20.99**	60.51
w/ Attention Sink	**99.05**	**96.92**	**63.85**	68.69	**100.00**	**78.21**	81.93	99.88	18.41	23.72	**23.28**	79.43	**98.39**	**99.16**	24.29	18.90	67.13
w/ Temporal Reindex	97.99	96.45	60.55	**70.74**	99.93	70.66	**86.75**	99.65	17.46	22.96	22.28	76.86	97.62	98.22	61.79	19.21	68.69
w/ Rolling Semantics	98.24	96.78	62.79	69.41	**100.00**	69.84	80.58	**99.99**	**19.05**	**24.82**	**23.28**	**82.86**	98.08	98.44	**70.71**	19.20	**69.63**
Sink/Cache Ratio: 5/6																	
No Sink Chunk	96.79	96.53	59.16	69.80	86.80	36.39	64.33	71.21	10.79	22.20	19.91	68.86	97.63	98.14	48.57	**20.99**	60.51
w/ Attention Sink	**98.98**	**97.62**	62.46	69.13	**100.00**	**70.00**	**87.32**	**100.00**	23.81	**25.11**	**23.60**	73.71	97.57	98.36	53.57	18.69	68.75
w/ Temporal Reindex	98.25	97.01	62.51	**70.39**	99.96	69.96	79.36	**100.00**	**26.03**	24.33	23.09	**80.00**	97.66	98.26	66.07	18.87	69.48
w/ Rolling Semantics	98.58	96.94	**63.08**	69.68	**100.00**	69.98	80.23	**100.00**	21.59	25.03	23.16	78.00	**98.16**	**98.65**	**74.69**	18.91	**69.79**

Table S3: **Full ablation on 5-minute AR video synthesis (all `VBench-Long` dimensions).** The longer 5-minute rollout widens the horizon gap, so the effect of each cache behavior is more visible than at 1 minute. The overall score rises as low-drift content, sliding temporal indices, and evolving semantics are added, with the best result at Rolling Sink, $S/K = 5/6$. **Bold** and underline mark the best and second-best within each ratio.

Cache Strategy	VBench-Long Dimensions ↑																Avg
	SubCon	BgCon	AesQl	ImgQl	ObjCls	MulObj	Col	SpaRel	Sce	TmpSty	OvrCon	HumAct	TmpFli	MoSmo	DyDg	AppSty	Score↑
Sink/Cache Ratio: 1/6																	
No Sink Chunk	94.24	**96.10**	42.89	57.01	33.39	14.27	61.05	35.70	4.30	11.32	11.39	27.10	98.20	98.65	24.19	20.53	45.65
w/ Attention Sink	**95.86**	95.71	50.37	61.93	52.54	23.53	52.78	45.78	5.73	14.25	16.36	44.52	96.94	**98.88**	16.94	20.48	49.54
w/ Temporal Reindex	95.14	95.71	49.61	61.36	61.47	31.87	63.49	68.28	7.89	13.90	16.92	48.39	98.22	97.31	**70.97**	20.50	56.31
w/ Rolling Semantics	95.35	95.23	**58.10**	**64.50**	**89.35**	**50.50**	**69.45**	**89.40**	**20.79**	**23.08**	**21.85**	**73.23**	**98.41**	98.05	54.84	**21.20**	**63.96**
Sink/Cache Ratio: 2/6																	
No Sink Chunk	94.24	**96.10**	42.89	57.01	33.39	14.27	61.05	35.70	4.30	11.32	11.39	27.10	98.20	98.65	24.19	20.53	45.65
w/ Attention Sink	**96.65**	95.91	54.91	**69.28**	81.67	43.63	72.87	68.50	8.96	15.65	17.59	65.48	98.19	**99.05**	6.05	**20.68**	57.19
w/ Temporal Reindex	95.87	95.30	51.93	62.81	81.07	39.58	68.26	80.62	7.17	16.20	17.93	65.48	97.90	97.58	64.52	20.55	60.17
w/ Rolling Semantics	96.45	95.66	**60.65**	65.38	**98.41**	**63.79**	**73.24**	**96.46**	**23.30**	**24.24**	**22.00**	**76.77**	**98.27**	98.35	**68.15**	20.58	**67.61**
Sink/Cache Ratio: 3/6																	
No Sink Chunk	94.24	**96.10**	42.89	57.01	33.39	14.27	61.05	35.70	4.30	11.32	11.39	27.10	98.20	98.65	24.19	**20.53**	45.65
w/ Attention Sink	**97.84**	95.92	58.05	**69.86**	95.85	**76.11**	70.07	96.03	14.70	19.59	20.11	77.10	**98.53**	**99.10**	14.11	20.18	63.95
w/ Temporal Reindex	96.10	95.13	56.57	68.79	93.71	51.75	76.67	89.70	11.47	19.43	19.99	65.48	97.62	97.03	**71.77**	20.28	64.47
w/ Rolling Semantics	96.79	95.97	**62.06**	67.05	**99.80**	68.87	**80.92**	**99.87**	**16.13**	**24.68**	**22.37**	**81.94**	98.28	98.36	67.34	20.02	**68.78**
Sink/Cache Ratio: 4/6																	
No Sink Chunk	94.24	**96.10**	42.89	57.01	33.39	14.27	61.05	35.70	4.30	11.32	11.39	27.10	98.20	98.65	24.19	**20.53**	45.65
w/ Attention Sink	**98.47**	96.02	62.10	69.51	**100.00**	**79.68**	76.37	99.80	**20.43**	22.58	22.19	77.74	**98.57**	**99.17**	27.42	19.06	66.82
w/ Temporal Reindex	96.46	95.13	59.01	**69.81**	88.59	65.91	80.76	94.79	**20.43**	22.12	21.78	78.71	97.67	97.55	67.74	19.56	67.25
w/ Rolling Semantics	97.01	96.07	**62.27**	68.80	**100.00**	69.68	**82.51**	**99.82**	17.20	**24.84**	**22.46**	**85.81**	98.25	98.34	**72.18**	19.73	**69.69**
Sink/Cache Ratio: 5/6																	
No Sink Chunk	94.24	96.10	42.89	57.01	33.39	14.27	61.05	35.70	4.30	11.32	11.39	27.10	98.20	**98.65**	24.19	**20.53**	45.65
w/ Attention Sink	**98.96**	**97.29**	62.70	69.18	**100.00**	70.00	**88.19**	**100.00**	22.94	25.12	**23.56**	73.23	97.58	98.36	62.90	18.64	69.29
w/ Temporal Reindex	97.27	96.03	61.98	**70.21**	99.17	70.40	82.92	99.72	**26.16**	24.45	22.90	80.97	97.65	97.91	62.50	18.95	69.32
w/ Rolling Semantics	98.04	96.29	**62.96**	69.87	**100.00**	**72.84**	78.83	**100.00**	**26.16**	**25.33**	23.10	**87.10**	**98.32**	98.59	**64.11**	18.91	**70.03**

with Attention Sink (Sec. 3.3.1), sliding temporal indices with Temporal Re-index (Sec. 3.3.2), and evolving semantics with Rolling Semantics (Sec. 3.3.3). Results are reported under different sink/cache ratios on both 1-minute (Tab. S2) and 5-minute (Tab. S3) AR video synthesis.

Across sink ratios and video lengths, adding each cache behavior raises the overall `VBench-Long` score (Huang et al., 2024; Zheng et al., 2025; Huang et al., 2025b), which supports jointly maintaining low-drift content, sliding temporal indices, and evolving semantics. We use $S/K = 5/6$ as the default because it scores best while still keeping one recent chunk as a local-motion anchor. This sweep varies S/K at the fixed budget $K = 6$ shared by both bases. Other budgets and chunk sizes would require differently-trained bases (Sec. 5).

D Discussion on Video Repetition

A large sink ratio raises a natural concern: does the model just repeat early content? Rolling Sink works against this in two ways. First, it keeps one recent chunk, which carries the current pose,



Figure S2: **Feature distance to the first frame over a long rollout.** We plot the normalized feature distance between each generated frame and the first frame. Exact replay of the opening content would drive this distance toward zero in periodic dips. Against LongLive (Yang et al., 2025), Rolling Sink shows no such collapses at the marked timestamps, which indicates the content keeps changing instead of replaying the start. This detects one repetition mode (return to the first frame). The bank-periodic replay and progression measure is in App. E. The marked timestamps are the frames shown in Fig. S3.

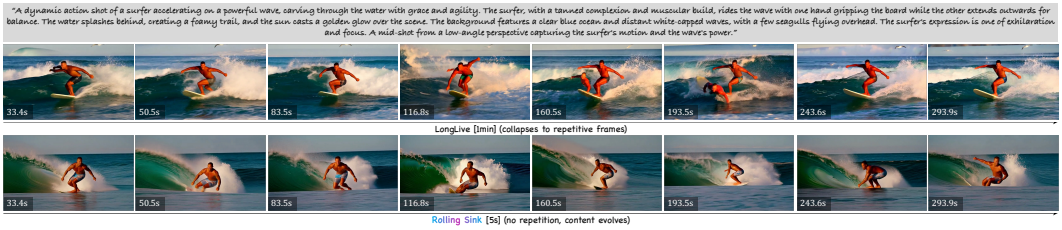

Figure S3: **Visual examples of content evolution.** We show frames at the timestamps marked in Fig. S2, which are candidate returns toward earlier visual states. Against LongLive (Yang et al., 2025), Rolling Sink holds identity, color, and scene structure while poses, motion, camera view, and local appearance keep changing across time, instead of directly replaying earlier frames.

motion phase, camera view, and local appearance, so the next prediction is anchored to the current rollout rather than to early sink content alone. Second, Rolling Semantics stops the sink from acting as a fixed prefix: it evolves the sink's semantics along a forward-reverse traversal of the low-drift bank and re-indexes the selected chunks to the current temporal window, so both the sink's semantics and positions move with the rollout.

Fig. S2 gives diagnostic evidence, comparing Rolling Sink with LongLive (Yang et al., 2025), a training-based baseline trained on 1-minute videos (Tab. 4). The distance to the first frame stays non-trivial over long rollouts, and the frames in Fig. S3 keep evolving while subject and scene structure stay stable. This diagnostic covers one repetition mode and is not a guarantee against periodicity. The bank of low-drift chunks is finite, so a long enough rollout can still read out periodically, and the forward-reverse traversal places a partial time-reversal on the sink. A direct bank-periodic replay and progression measure is given in App. E. How the forward-reverse schedule affects irreversible motions (an object falling, liquid pouring, something breaking or being consumed) is untested here: we run no with/without Rolling Semantics comparison on such motions and no human study, and mark both open (Sec. 5). The large sink ratio should therefore be read as preserving low-drift cache quality, not as a proof of unbounded content progression.

E Periodic Replay and Content Progression

The finite low-drift bank raises a concern that Rolling Sink could periodically replay or reverse earlier content over long rollouts, rather than letting content progress (Sec. 5). App. D tracks the distance to the first frame. Here we measure self-similarity at multiples of the bank length and calibrate it against a bank-loop control.

Setup. On five 5-minute rollouts we extract a per-frame feature (DINOv2 ViT-S/14 (Oquab et al., 2023) and CLIP ViT-B/32 (Radford et al., 2021) image embedding, ℓ_2-normalized) and compute the mean cosine distance at time lag τ, $d(\tau) = \mathrm{mean}_t\,[\,1 - \langle \boldsymbol{f}_t, \boldsymbol{f}_{t-\tau} \rangle\,]$. The cache bank spans $K = 6$ chunks (48 frames) and the forward-reverse schedule completes a cycle every two banks, so replay-

Table S4: **Self-similarity at the bank period vs. off-bank lags** (mean cosine distance over five 5-minute rollouts, larger is more distinct, and dip $= d_{\text{off-bank}} - d_{\text{bank}}$, where a positive dip demonstrates bank-periodic replay). A bank loop collapses to 0 at every bank-multiple lag (large dip), whereas Rolling Sink is uniformly distinct across bank and off-bank lags (dip ≈ 0), so it neither replays nor reverses the finite bank.

Feature	Condition	bank lags	off-bank lags	dip
DINOv2 ViT-S/14	Rolling Sink	0.043	0.043	−0.00
	Bank Loop	0.000	0.050	**+0.05**
CLIP ViT-B/32	Rolling Sink	0.021	0.020	−0.00
	Bank Loop	0.000	0.023	**+0.02**

Table S5: **Abbreviation legend for the `VBench-Long` dimensions** reported in the quantitative tables, listed in column order.

Abbr.	Dimension	Abbr.	Dimension
SubCon	subject consistency	Sce	scene
BgCon	background consistency	TmpSty	temporal style
AesQl	aesthetic quality	OvrCon	overall consistency
ImgQl	imaging quality	HumAct	human action
ObjCls	object class	TmpFli	temporal flickering
MulObj	multiple objects	MoSmo	motion smoothness
Col	color	DyDg	dynamic degree
SpaRel	spatial relationship	AppSty	appearance style

ing or oscillating over the bank would make d dip at bank-multiple lags ($\tau \in \{48, 96, 144, 192\}$) relative to off-bank lags ($\tau \in \{24, 72, 120, 168\}$). A flat profile (no dip) therefore indicates no measurable bank-periodic replay or reversal. As a control we tile the first bank (48 frames) to full length, a bank-periodic reference.

Result. As Tab. S4 shows, the bank-loop control collapses to zero distance at every bank-multiple lag while staying high off-bank (dip $+0.05$ for DINOv2, $+0.02$ for CLIP), the signature of replaying the bank. Rolling Sink shows no such dip: its distance at bank-multiple lags matches that at off-bank lags (dip ≈ 0) under both features, so it neither replays the bank nor settles into the two-bank period of the forward-reverse schedule. Together with the sustained distance to the opening frames in App. D, this indicates the rollout keeps evolving rather than recycling the finite bank. This is a global-content periodicity measure. It does not by itself certify behavior on individual irreversible events (an object falling, liquid pouring, something breaking or being consumed), and long-range novelty is ultimately bounded by the base model's training. We mark both open in Sec. 5.

F Algorithm Pseudo-code

Alg. S1 summarizes open-ended AR generation with Rolling Sink. The model first generates the initial K chunks within the training horizon, which serve as a low-drift chunk bank. After the rollout exceeds the training horizon, Rolling Sink builds the cache at each step by selecting sink chunks from a forward-reverse traversal of the low-drift chunk bank, re-indexing them to the current temporal window, and concatenating them with the recent chunks. Here, REVERSE flips the latent-frame order within a chunk (Eq. 6 and 7), and REINDEX applies the RoPE temporal index to each latent frame (Eq. 5).

G `VBench-Long` Evaluation Protocol

`VBench-Long` (Huang et al., 2024; 2025b; Zheng et al., 2025) scores a long video on 16 dimensions, each read out by a separate pretrained expert and all higher-is-better. Our quantitative tables (Tab. 2, 3, 4, S2, and S3) report every dimension under the abbreviations of Tab. S5.

Prompt suite. Each dimension ships with 70 to 100 prompts. We draw a fixed subset of 10 prompts per dimension once and reuse it for every quantitative result in the paper.

Algorithm S1 Open-Ended AR Generation with Rolling Sink

Require: Base AR video diffusion model p_θ; cache budget K; sink size S; chunk size n; total rollout steps N.
Ensure: Generated chunks $\boldsymbol{x}_{[0,N)}$.

// Generate low-drift chunks within the training horizon
for $i = 0$ **to** $K - 1$ **do**
    $\mathcal{C}_i \leftarrow \boldsymbol{x}_{[0,i)}$
    Sample $\boldsymbol{x}_i \sim p_\theta(\boldsymbol{x}_i \mid \mathcal{C}_i)$
end for

// Open-ended rollout with Rolling Sink
for $i = K$ **to** $N - 1$ **do**
    $\mathcal{C}_{\text{sink}} \leftarrow \emptyset$
    for $j = i - K$ **to** $i - (K - S) - 1$ **do**
        $q \leftarrow \lfloor j/K \rfloor, \quad r \leftarrow j \bmod K$
        if $q \bmod 2 = 0$ **then**
            $\boldsymbol{s} \leftarrow \boldsymbol{x}_r$ ▷ Forward rolling semantics
        else
            $\boldsymbol{s} \leftarrow \textsc{Reverse}(\boldsymbol{x}_{K-1-r})$ ▷ Reverse rolling semantics
        end if
        $\boldsymbol{s}^j \leftarrow \textsc{ReIndex}(\boldsymbol{s}, j, n)$
        $\mathcal{C}_{\text{sink}} \leftarrow \textsc{Append}\big(\mathcal{C}_{\text{sink}}, \boldsymbol{s}^j\big)$
    end for
    $\mathcal{C}_{\text{recent}} \leftarrow \boldsymbol{x}_{[i-(K-S),i)}$
    $\mathcal{C}_i \leftarrow \textsc{Cat}(\mathcal{C}_{\text{sink}}, \mathcal{C}_{\text{recent}})$
    Sample $\boldsymbol{x}_i \sim p_\theta(\boldsymbol{x}_i \mid \mathcal{C}_i)$
end for
return $\boldsymbol{x}_{[0,N)}$