

Surprise Forcing: What to Remember, When to Skip in Long Video Generation

Shuwei Shi¹, Zhen Li², Muyao Niu¹, Chuanhao Li², Bo Zheng², Kaipeng Zhang^{2,†}, and Yinqiang Zheng^{1,†}

¹ The University of Tokyo

shishuwei666@gmail.com, muyao.niu@gmail.com, yqzheng@ai.u-tokyo.ac.jp

² Alaya Lab

kp_zhang@foxmail.com

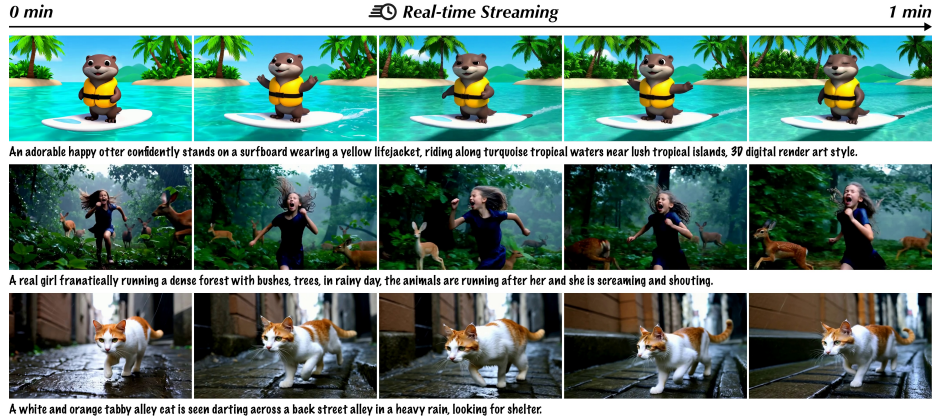


Fig. 1: Samples generated by Surprise Forcing. Our method can perform real-time long video generation while maintaining good temporal consistency.

Abstract. Autoregressive diffusion models have shown remarkable progress in long video generation by producing frames causally. To bound memory and computational cost, these models rely on a fixed-size sliding-window KV cache. While attention sinks retain a few initial frames as temporal anchors, the vast majority of historical context is irrecoverably discarded as the window advances, severing long-range dependencies essential for scene consistency and narrative coherence. Meanwhile, the fixed multi-step denoising schedule applies uniform effort to every chunk, ignoring the varying demands imposed by diverse scene dynamics across temporal segments. In this paper, we propose **Surprise Forcing**, a training-free framework that addresses both bottlenecks. We introduce a **Surprise-Gated Memory Bank** that scores each evicted frame by

[†] Corresponding authors.

its aggregate divergence from the bank and nearest-neighbor novelty, admitting only the most informative frames via a budget-normalized gate that adapts to video dynamics. A priority-based eviction policy keeps the bank compact yet informative, while dynamic routing retrieves the most relevant historical frames at attention time. We further propose **Surprise-Aware Denoising**, which uses intra-chunk cosine distance among adjacent frames after a single denoising step as a self-referential difficulty predictor, enabling adaptive step reduction for easy chunks while preserving full refinement for complex ones. Experiments on VBench show that Surprise Forcing maintains robust narrative coherence in long videos without compromising real-time responsiveness.

Keywords: Long Video Generation · Surprise-Gated Memory Bank · Surprise-Aware Denoising

1 Introduction

Diffusion-based text-to-video models [7, 27, 29, 31, 32, 38] have rapidly improved visual fidelity and prompt alignment, making long-form generation increasingly relevant for storytelling, interactive creation, and world simulation. To move beyond a few-second horizon under practical GPU constraints, recent autoregressive (AR) diffusion frameworks [4, 8, 9, 20, 37] generate videos *chunk-by-chunk* with a rolling key-value (KV) cache, often combined with distillation from a short-horizon bidirectional teacher to enable few-step causal sampling [40, 41]. While this paradigm makes long video generation feasible, *truly long* and *efficient* streaming generation still faces two fundamental bottlenecks in deployed systems: **memory** and **compute**.

Memory bottleneck. The standard rolling KV cache behaves like a FIFO buffer [9]: as the context window advances, older frames are irrevocably discarded. This bounded-memory design is computationally attractive, yet it systematically removes long-range cues that are essential for identity preservation, scene persistence, and narrative coherence, leading to drift and inconsistency over extended horizons. Prior mitigations lack the ability to adaptively manage valuable historical information, making it difficult to determine which frames should be preserved as the story evolves [5, 37, 39].

Compute bottleneck. AR diffusion generation is dominated by iterative denoising: each chunk typically runs a fixed multi-step schedule regardless of content. However, long videos are intrinsically non-homogeneous [24]: static backgrounds and temporally smooth segments often converge quickly, whereas chunks containing rapid motion, new objects, or scene transitions require substantially more refinement. Applying a uniform schedule therefore wastes a large fraction of inference compute on already-converged chunks, limiting throughput and responsiveness.

To mitigate the above issues, we propose **Surprise Forcing**, a *training-free* framework that improves both long-horizon consistency and efficiency by making *content-adaptive* decisions about **what to remember** and **when to skip**.

Our key insight is that a model can use its *own internal signals* to estimate informativeness and difficulty: only frames that are *surprising* should be committed to long-term memory, and only chunks predicted to be *hard* (i.e., with high surprise) should receive full denoising compute.

Surprise Forcing introduces two complementary inference-time mechanisms. First, a **Surprise-Gated Memory Bank** stores a compact set of informative historical frames that would otherwise be evicted from the rolling KV-cache window. We quantify frame informativeness using a dual-component surprise score that balances global deviation from existing memory and local nearest-neighbor novelty, and we maintain memory utility via content-adaptive routing and priority-based eviction. Second, a **Surprise-Aware Denoising** policy predicts per-chunk denoising difficulty after a single step using an intra-chunk surprise signal, and adaptively assigns either a reduced or full step schedule, skipping redundant refinement on easy chunks.

Extensive experiments on VBench [10] demonstrate that Surprise Forcing improves long-range consistency and overall quality while reducing inference overhead compared to existing autoregressive baselines, offering a practical path toward long-horizon, real-time streaming video generation, as shown in Fig. 1.

Our contributions can be summarized as follows:

- We propose a training-free surprise-driven framework, Surprise Forcing, which generates long-term consistent videos while maintaining the efficiency required for real-time interaction.
- We introduce a Surprise-Gated Memory Bank that selectively preserves semantically critical frames evicted from the sliding window KV cache using a dual-component surprise measure.
- We develop an eviction mechanism that preserves memory diversity and utility under a strict capacity budget.
- We propose Surprise-Aware Denoising to accelerate inference by using intra-chunk surprise to skip redundant denoising iterations for easy chunks.

2 Related Work

2.1 Autoregressive Long Video Generation

Diffusion has become a dominant paradigm for video synthesis, but naively scaling bidirectional diffusion over long sequences is computationally prohibitive. Training-free length extension methods reschedule noise or re-balance temporal frequencies to stretch pretrained models beyond their nominal horizon, including FreeNoise [25], FreeLong [21], and FreeLong++ [22], as well as approaches that support extremely long rollouts without additional training, such as FIFO-Diffusion [15] and rolling-style sampling schemes [26]. A complementary thread blends diffusion with causal prediction to stabilize variable-horizon conditioning, e.g., Diffusion Forcing [2] and HistoryGuidance [28], which are also adopted by industrial long-video systems (e.g., SkyReels-V2 [3] and Magi-1 [30]). More

recently, streaming AR diffusion systems explicitly target real-time or minute-scale generation: StreamingT2V [6] augments memory with randomized blending; CausVid [42] distills a bidirectional teacher into a few-step causal generator; StreamDiT [16] combines distillation with a moving buffer; Rolling Forcing [20] explores diffusion-forcing-style training for real-time AR rollouts; and Self Forcing [9] aligns training with inference by simulating causal rollout, with extensions such as Self-Forcing++ [4] improving long-horizon scalability via extended distillation formulations. These works primarily advance training and streaming rollout strategies; our method is orthogonal and focuses on *training-free* inference-time allocation of memory and compute.

2.2 Context Management in Long Video Generation

Long-horizon generation requires persistent access to historical state, but attention cost scales poorly with context length. Most streaming systems therefore adopt local-window attention with a bounded KV cache. To mitigate instability, attention sinks preserve a small set of initial frames as anchors, originally studied in StreamingLLM [35] and later adopted in streaming video generation pipelines. Nevertheless, the cache window still implies hard forgetting of distant history, motivating improved context management. Closest to the training-free streaming setting, Deep Forcing [39] revisits attention-sink based KV-cache management through deep sink and participative compression, preserving high-participation tokens within the bounded sliding window. Surprise Forcing is complementary: instead of pruning token-level context inside the window, it preserves selected evicted frames in a compact external bank and retrieves frame-level memories according to surprise and query relevance. Beyond such in-window KV management, one line of work focuses on **context compression**. LoViC [13] compresses long contexts to enable efficient long generation. FramePack-style designs pack input frame context for AR prediction [44], and recent work studies explicit memory encoders trained for frame reconstruction and retrieval to balance the context-length and quality trade-off [45]. Pack-and-force strategies further enforce long-form consistency under short-context constraints [33]. Another direction is **retrieval and explicit memory**. This retrieval-and-memory view is conceptually related to retrieval-augmented generation (RAG) in LLMs, where a generator is conditioned on explicit non-parametric memory retrieved from an external index [17]. In long video generation, however, the retrieved units are temporally ordered visual states rather than text passages, and retrieval must remain causal and bounded by the streaming KV-cache budget. WorldMem [36] and Context-as-Memory [43] retrieve relevant history for long-horizon world simulation and interactive generation; VMem [18] indexes past views via surfels to recover relevant viewpoints; and Memory Forcing [8] couples geometry-indexed memory with tailored training regimes to balance exploration and revisits. MemFlow [12] proposes adaptive memory mechanisms for consistent long video narratives under prompt transitions. Compared to trained memory systems and in-window KV compression methods, Surprise Forcing provides a lightweight,

training-free external memory bank that prioritizes evicted historical frames by surprise and routes retrieval dynamically during streaming inference.

2.3 Efficient Video Generation

Efficiency is a central requirement for long streaming video generation. A major line of work reduces diffusion steps via distillation, including Distribution Matching Distillation (DMD) [41] and improved variants [40], enabling few-step or even one-step synthesis for images and serving as a foundation for fast video distillation pipelines. Streaming AR diffusion systems like CausVid [42] and StreamDiT [16] also incorporate multi-step distillation to improve throughput. Training-free caching provides another complementary acceleration route. Ada-Cache [14] accelerates video Diffusion Transformers (DiTs) by reusing Transformer-block residuals across denoising steps and uses motion regularization to adapt computation according to video motion content. Our Surprise-Aware Denoising shares the high-level goal of content-adaptive compute allocation, but differs in operating at the chunk-level during streaming AR generation: after the first denoising step, it uses intra-chunk surprise to decide whether intermediate denoising steps can be skipped.

Another line accelerates the transformer backbone by reducing attention cost or token count. Sparse or structured attention has been explored for long video generation, such as Radial Attention [19], Sparse VideoGen [34], and mixture-based designs like Mixture-of-Contexts [1] that dynamically select relevant context for attention computation. These methods primarily target architectural efficiency. In contrast, Surprise Forcing focuses on adaptive allocation: it keeps the backbone and trained weights fixed, while dynamically deciding which history to store or retrieve and how much denoising compute to spend on each chunk during a long rollout.

3 Method

Our proposed framework is built upon the Self Forcing paradigm for autoregressive long video generation. In Sec. 3.1, we first provide the preliminaries of this paradigm, while further elaborating on the underlying mechanism of streaming long-video tuning. Sec. 3.2 presents a high-level overview of our proposed framework. In Sec. 3.3, we detail the Surprise-Gated Memory Bank, designed to preserve semantically critical historical information. Finally, we introduce the Surprise-Aware Denoising strategy in Sec. 3.4, which dynamically allocates computational resources according to the inherent complexity of video chunks.

3.1 Preliminaries

We consider streaming autoregressive video diffusion that generates a long video causally in fixed-length chunks. Given text condition c , the AR factorization is

$$p_{\theta}(\mathbf{x}_{1:N} \mid c) = \prod_{n=1}^N p_{\theta}(\mathbf{x}_n \mid \mathbf{x}_{<n}, c), \quad (1)$$

where $\mathbf{x}_n$ denotes the n -th generated latent chunk and $\mathbf{x}_{<n}$ is represented by a fixed-size rolling KV cache.

To obtain an efficient few-step generator, we adopt Distribution Matching Distillation (DMD) to align a streaming student with a short-horizon bidirectional teacher. Specifically, t denotes a diffusion time step sampled from the training schedule, and $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ is the generator noise. $G_\theta(\mathbf{z})$ denotes the student generator output (conditioned on c and the rolling cache). $\Phi(\cdot, t)$ denotes the forward noising operator at step t (used for backward-noise initialization on self-rollback windows). $p_{\theta,t}^S$ and p_t^T are the student/teacher distributions at diffusion step t , and their scores are $s_\theta^S(\mathbf{x}_t, t) = \nabla_{\mathbf{x}_t} \log p_{\theta,t}^S(\mathbf{x}_t)$ and $s^T(\mathbf{x}_t, t) = \nabla_{\mathbf{x}_t} \log p_t^T(\mathbf{x}_t)$, respectively. The DMD objective is:

$$\mathcal{L}_{\text{DMD}} = \mathbb{E}_t [\text{KL}(p_{\theta,t}^S \| p_t^T)], \quad (2)$$

and its score-form gradient:

$$\nabla_\theta \mathcal{L}_{\text{DMD}} \approx -\mathbb{E}_{t,\mathbf{z}} \left[\int \left(s^T(\Phi(G_\theta(\mathbf{z}), t), t) - s_\theta^S(\Phi(G_\theta(\mathbf{z}), t), t) \right) \frac{\partial G_\theta(\mathbf{z})}{\partial \theta} d\mathbf{z} \right]. \quad (3)$$

Since the teacher is only reliable within a short temporal horizon, we extend DMD to long videos by applying the same KL objective to uniformly sampled windows from a long self-rollback under the rolling cache. Concretely, we roll out for N chunks, sample $i \sim \text{Unif}\{1, \dots, N - K + 1\}$, and take a contiguous window $W_i := \mathbf{x}_{i:i+K-1}$ of length K (matching the teacher horizon), which is re-noised by $\Phi(\cdot, t)$ before computing:

$$\mathcal{L}_{\text{DMD}}^{\text{ext}} = \mathbb{E}_t \mathbb{E}_{i \sim \text{Unif}\{1, \dots, N - K + 1\}} [\text{KL}(p_{\theta,t}^S(W_i) \| p_t^T(W_i))]. \quad (4)$$

3.2 Overview

We propose **Surprise Forcing**, a framework built upon two core contributions, as shown in Fig. 2. First, we introduce a **Surprise-Gated Memory Bank** (Sec. 3.3) that selectively preserves the most informative evicted frames from cache in a compact external memory, coupled with a dynamic routing mechanism that retrieves the most relevant historical frames at attention time. Second, we propose **Surprise-Aware Denoising** (Sec. 3.4) that leverages intra-chunk surprise signals to predict per-chunk denoising difficulty after a single step, adaptively reducing the number of denoising steps for easy chunks while preserving full refinement for complex ones.

3.3 Surprise-Gated Memory Bank

Not all frames carry equal information: static or repetitive segments contribute little to long-range coherence, whereas frames depicting scene transitions, novel objects, or significant motion changes are critical to preserve. To quantify the informativeness of each evicted frame and decide whether to admit it into the

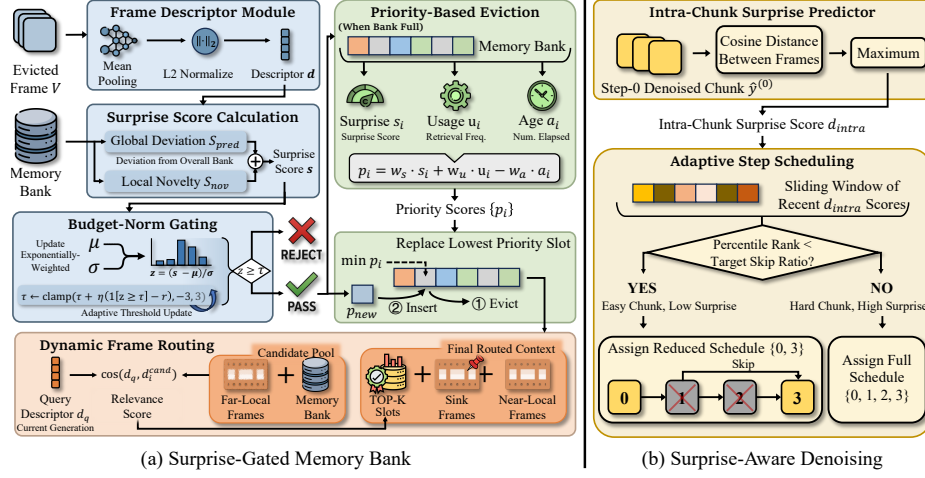


Fig. 2: The framework of Surprise Forcing. Our method consists of two surprise-driven components. The first is the Surprise-Gated Memory Bank, which regulates the writing and eviction of memory entries based on surprise-related signals. The second is Surprise-Aware Denoising, which employs an Intra-Chunk Surprise Predictor to estimate whether the denoising process for the current chunk can be accelerated.

bank, we define a surprise metric based on its novelty relative to the current bank contents, coupled with a budget-controlled gating mechanism.

Frame Descriptor. Given an evicted frame with value tokens $\mathbf{V} \in \mathbb{R}^{L \times D}$ (where L is the number of spatial tokens and D is the feature dimension), we compute a compact descriptor by mean-pooling over the spatial dimension followed by ℓ_2 normalization:

$$\bar{\mathbf{v}} = \frac{1}{L} \sum_{l=1}^L \mathbf{v}_l \in \mathbb{R}^D, \quad \mathbf{d} = \frac{\bar{\mathbf{v}}}{\|\bar{\mathbf{v}}\|_2}. \quad (5)$$

We use value tokens rather than key tokens for descriptor computation for two reasons. First, key representations in our architecture are applied with rotary position embeddings (RoPE) at attention time; although raw keys are stored without RoPE, the key projection is co-trained with position-dependent objectives, causing implicit positional biases that interfere with content-based similarity. Second, keys are optimized for discriminative query-key dot products in attention, which encourages head-specific routing patterns rather than faithful frame-content representation. In contrast, value tokens carry the information propagated through the attention output and are trained to preserve semantic content, making them more suitable for content descriptors.

Surprise Score. A natural approach to measuring frame novelty would be a single similarity metric against the bank. However, using only the average similarity to all bank entries fails to detect cases where a frame is globally dissimilar but nearly duplicates one specific bank entry. Conversely, relying solely on

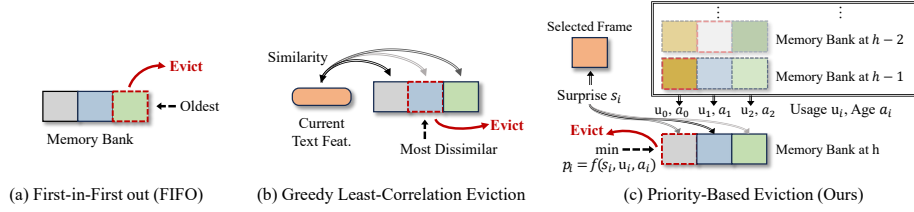


Fig. 3: Comparison of historical context eviction strategies. Unlike previous methods that rely solely on temporal order or current relevance as the eviction criterion, our approach provides a more principled mechanism for memory eviction.

the nearest-neighbor similarity ignores the overall distributional novelty, since a frame may differ from its closest match yet still be well-represented by the bank as a whole. To capture both aspects, we define two complementary measures. Let $\{\mathbf{d}_i\}_{i=1}^M$ denote the descriptors of the M frames currently stored in the bank. We compute $s_{\text{pred}} = \frac{1}{2}(1 - \frac{1}{M} \sum_{i=1}^M \cos(\mathbf{d}, \mathbf{d}_i))$ as the global prediction error, which measures how much the frame deviates from the bank’s overall representation, and $s_{\text{nov}} = \frac{1}{2}(1 - \max_i \cos(\mathbf{d}, \mathbf{d}_i))$ as the local novelty, which measures whether the frame introduces genuinely new content not covered by any existing entry. The overall surprise score is their weighted combination:

$$s = \alpha \cdot s_{\text{pred}} + (1 - \alpha) \cdot s_{\text{nov}} \quad (6)$$

Budget-Norm Gating. A fixed statistical threshold for surprise-based gating would be either too permissive or too restrictive across different video contents—action sequences produce consistently high surprise while static scenes yield uniformly low values. To achieve content-adaptive write control, we normalize the raw surprise score to a z-score $z = (s - \mu)/\sigma$ using exponentially-weighted running estimates of the mean and variance. The detailed procedure is provided in the supplementary material. A frame is accepted when $z \geq \tau$, where the threshold τ is updated after each decision via $\tau \leftarrow \text{clamp}(\tau + \eta \cdot (\mathbf{1}[z \geq \tau] - r), -3, 3)$ with learning rate η and target write rate r . This feedback loop drives the write rate to converge to r regardless of the absolute surprise scale. When a frame is accepted ($\mathbf{1}[z \geq \tau] = 1$), the threshold increases by $\eta(1 - r)$, raising the bar for subsequent frames; when a frame is rejected ($\mathbf{1}[z \geq \tau] = 0$), the threshold decreases by ηr , lowering the bar. At equilibrium, the acceptance rate converges to r . We clamp τ to $[-3, 3]$ because z-scores beyond this range correspond to the extreme tails of the distribution (beyond 99.7% under a Gaussian assumption): $\tau < -3$ would accept virtually all frames, while $\tau > 3$ would reject virtually all, rendering the gate ineffective. The target write rate r provides an explicit control over the trade-off between bank freshness and stability. A bank with limited capacity must balance two competing needs: updating frequently enough to track evolving video content, and retaining entries long enough for them to be useful during routing. More details can be found in the supplementary material.

Priority-Based Eviction. When the memory bank reaches its capacity C and a new frame passes the surprise gate, an existing slot must be replaced. Simple eviction strategies, such as removing the oldest entry or the entry that appears least useful for the current context, would be suboptimal. The oldest frame may still be frequently retrieved by the routing mechanism, and a frame that seems currently uninformative may have become the sole representative of a particular scene segment after other similar entries were evicted, as shown in Fig. 3. Effective eviction therefore requires balancing multiple competing objectives: retaining frames that are still informative, frequently accessed, and non-redundant, while removing those that have become stale or duplicated.

We formulate eviction as a composite priority scoring problem. For each bank slot i , we compute a priority score that aggregates three normalized signals, all normalized to $[0, 1]$: the surprise score s_i recorded at write time reflecting initial informativeness, a usage score u_i tracking retrieval frequency via exponentially-decayed soft-relevance updates, an age penalty a_i counting how many chunks have elapsed since the frame was written:

$$p_i = w_s \cdot s_i + w_u \cdot u_i - w_a \cdot a_i \quad (7)$$

The pending frame computes a corresponding priority p_{new} using the same formulation. Eviction occurs when $p_{\text{new}} > \min_i p_i$, replacing the lowest-priority slot. Without the usage signal, informative but infrequently-accessed frames would persist indefinitely; without the age penalty, early-generation frames would never yield their slots to more recent content.

Dynamic Frame Routing. The surprise-gated write path determines what to store in the memory bank, but an equally important question is what to retrieve at attention time. Naively attending to all bank entries would linearly increase the attention cost with bank size and dilute attention over irrelevant history. Conversely, using a fixed retrieval pattern (*e.g.*, always selecting the most recent bank entries) would defeat the purpose of content-aware storage.

To efficiently retrieve the most relevant historical context, we propose a dynamic frame routing mechanism that scores each candidate based on its semantic relevance to the current generation query. For each chunk, we construct a candidate pool by concatenating all valid bank frames with far-local frames, which are frames in the sliding-window cache that fall outside the near-local neighborhood but have not yet been evicted. Given the current query descriptor $\mathbf{d}_q$, we compute a cosine-similarity relevance score $\text{score}_i = \cos(\mathbf{d}_q, \mathbf{d}_i^{\text{cand}})$ for each candidate and select the top- k candidates to participate in the attention computation alongside the sink and near-local frames. This content-adaptive retrieval ensures that the limited global attention budget is allocated to the most informative historical frames, while the candidate pool combining bank and far-local frames provides temporal coverage spanning the entire generation history.

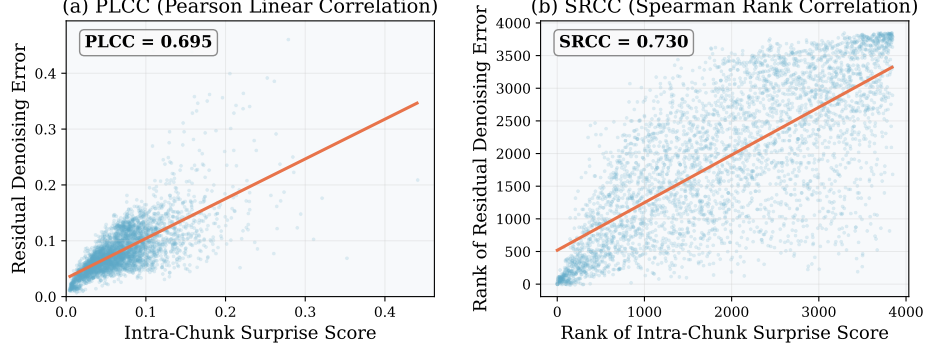


Fig. 4: The statistical correlation between the intra-chunk surprise score and the MSE between the first and the final denoising steps of each chunk. As illustrated in (a) and (b), the two variables exhibit a clear positive relationship, which is visually evident from the strong upward trend in both plots.

3.4 Surprise-Aware Denoising

In autoregressive video generation with the Surprise Forcing framework, each chunk undergoes a multi-step iterative denoising process. With a standard 4-step schedule, each of the chunks requires four full forward passes through the transformer, making the denoising loop the dominant computational bottleneck. However, not all chunks require the same amount of iterative refinement: we find that chunks depicting smooth temporal transitions or continuing a stable scene (low surprise) often converge to their final output after just one or two denoising steps, while chunks with rapid motion or scene changes (high surprise) require all four steps for faithful generation. Applying a uniform schedule to all chunks, therefore wastes substantial computation on easy chunks while providing limited benefit to difficult chunks.

Intra-Chunk Surprise Predictor. We observe that a lightweight yet effective surprise signal is available after just the *first* denoising step: the surprise score within the step-0 output itself. After the first denoising step (t_0), each chunk produces a partially denoised output $\hat{y}^{(0)} \in \mathbb{R}^{B \times T \times C \times H \times W}$ consisting of T frames. We find that the surprise score between adjacent frames within this output strongly correlates with the remaining denoising difficulty. Intuitively, if the step-0 output already exhibits large inter-frame displacements, the underlying scene dynamics are complex and subsequent steps are needed; if frames are nearly static, the chunk has essentially converged.

To quantify the intra-chunk surprise score, we compute the cosine distance between the flattened latent representations of adjacent frames and take the maximum:

$$c_t = 1 - \cos(\text{vec}(\hat{y}_t^{(0)}), \text{vec}(\hat{y}_{t+1}^{(0)})), \quad t = 1, \dots, T-1 \quad (8)$$

$$d_{\text{intra}} = \max_t c_t \quad (9)$$

where $\text{vec}(\cdot)$ denotes flattening the $C \times H \times W$ spatial dimensions into a single vector. As shown in Fig. 4, we randomly sample 50 prompts from the MovieGen [23] dataset and generate 1-minute videos for each prompt. For every chunk, we compute the intra-chunk surprise score as well as the MSE between the first and the final denoising steps of that chunk. We then analyze the correlation between these two quantities using both PLCC and SRCC. The results show that the correlation reaches approximately 0.7, indicating a strong positive relationship between the intra-chunk surprise score and the difficulty of denoising. This observation suggests that the intra-chunk surprise score can serve as an effective signal for determining whether denoising steps can be safely skipped. We use the maximum rather than the mean because denoising difficulty is dominated by *peak* motion: a chunk with one fast-moving frame and two static frames still requires full refinement to faithfully resolve the dynamic frame. We use cosine distance rather than L_2 distance because it is scale-invariant, measuring directional change in the latent space without being affected by the global magnitude of the denoised output, which varies across timesteps and video contents. Notably, this signal is *self-referential*: it is computed solely from the current chunk’s step-0 output and does not depend on previous chunks. This makes it free from cold-start issues and robust across different model checkpoints.

Adaptive Step Scheduling. Given the intra-chunk surprise score d_{intra} , we adaptively assign a reduced step schedule to easy chunks while preserving the full schedule for hard ones. Rather than using a fixed threshold, which would suffer from the same content-sensitivity issues discussed in Sec. 3.3, we rank each chunk’s d_{intra} within a sliding window of recent chunks and allocate steps based on its percentile rank:

$$\text{steps} = \begin{cases} \{0, 3\} & \text{if rank} < r_{\text{skip}} \quad (\text{skip intermediate steps}) \\ \{0, 1, 2, 3\} & \text{otherwise} \quad (\text{full schedule}) \end{cases} \quad (10)$$

where r_{skip} is the target skip ratio controlling the fraction of chunks that receive reduced schedules. The sliding window makes the percentile ranking adaptive to local video dynamics: a chunk that would be “easy” in an action scene may rank as “hard” in a static segment. The first several chunks are always run with the full schedule as a warmup to populate the ranking window.

4 Experiments

4.1 Implementation Details

Baseline model. We build our method on LongLive [37], which extends short video generation models to long-form generation through streaming training and inference. Specifically, LongLive is built upon Wan2.1-T2V-1.3B [31] and converts the original model into a causal-attention video generator using a Self Forcing DMD pipeline, enabling autoregressive generation over video clips. During training and inference, the model generates videos in a streaming manner by sequentially producing short clips and conditioning on previously generated

Table 1: Quantitative comparisons on VBench(5s) [10] short videos and VBench-Long(60s) [11] long videos. Under comparable inference speed, our method achieves state-of-the-art performance across all metrics, and the speedup becomes more pronounced when generating longer videos.

Model	VBench				VBench-Long			
	Total↑	Sub. Consistency ↑	Mul. Objects ↑	FPS ↑	Quality↑	Sub. Consistency ↑	Aesthetic ↑	FPS ↑
CausVid [42]	80.47	94.66	79.99	15.84	75.97	96.41	48.42	15.84
Self Forcing [9]	82.12	96.23	80.22	15.84	80.53	97.19	55.98	15.84
Rolling Forcing [20]	80.49	96.21	84.53	15.52	80.23	98.3	60.97	15.52
LongLive [37]	80.63	96.19	84.89	18.01	80.32	98.42	61.71	18.01
Surprise Forcing	83.40	96.28	87.50	16.45	81.88	98.51	63.98	17.18

content, allowing the total generation length to extend up to 60 seconds while maintaining temporal coherence.

Configurations. For long video generation, we set the bank capacity C to 6 to balance computational efficiency and the size of stored historical memory. For short videos, we set C to 3 to better match their shorter temporal context. We set the surprise mixing weight $\alpha = 0.7$ to balance global prediction error s_{pred} and local novelty s_{nov} , and use budget-norm gating with EMA momentum $m = 0.95$ to adapt the z-threshold τ (update step size $\eta = 0.1$) towards a target write rate $r = 0.3$; τ is clamped to $[-3, 3]$. For priority-based eviction, we use weights $\{w_s, w_u, w_a\} = \{1.8, 1.0, 0.4\}$ over the surprise, usage, and age terms, and dynamic routing retrieves the top- k bank frames with $k = 3$. We warm up the bank by accepting all frames in the first 3 chunks and then resetting the stored surprise scores to the current EMA mean μ ; for surprise-aware denoising, chunk difficulty is ranked within a sliding window of 20 recent chunks, and the first 5 chunks always use the full denoising schedule to bootstrap the ranking window. We set $r = 0.3$ to ensure that roughly 30% of evicted frames are admitted, keeping the bank responsive to scene changes without excessive churn that would destabilize the stored representations.

Evaluation Metrics. We adopt the metrics from VBench and VBench-Long, along with their 946 prompts, to evaluate the generated 5s and 60s videos, respectively. We report the overall Total score on VBench and the overall Quality score on VBench-Long, each accompanied by representative Semantic and Quality metrics. Following Self Forcing [9], we also report the throughput (FPS). All methods generate videos at a resolution of 832×480 .

4.2 Comparisons with Existing Alternatives

We compare Surprise Forcing with several representative methods. CausVid [42] is the first to distill a bidirectional teacher model into an autoregressive diffusion model using DMD. Building upon this, Self Forcing [9] introduces a self-rollout strategy to bridge the train-test gap. Rolling-Forcing [20] and LongLive [37] further extend this framework by adopting specific noise schedules or context management mechanisms to enable long video generation.

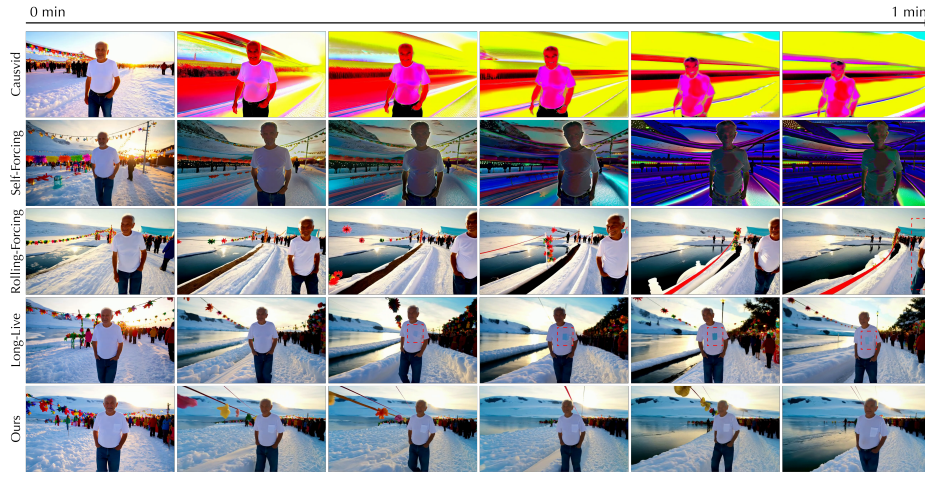


Fig. 5: Qualitative comparison with other methods. Our approach maintains strong subject consistency and smooth motion continuity during long video generation, while introducing minimal degradation in visual quality.

Quantitative Results. We conduct quantitative evaluations on VBench [10] and VBench-Long [11] using the official prompts provided by the benchmarks, generating videos of 5 seconds and 60 seconds respectively. We also measure the inference speed of each method on a single GPU, reported using the throughput metric. As shown in Tab. 1, our method significantly outperforms other approaches on the 5-second benchmark, achieving a 3.1% improvement in the Multiple Objects, indicating that the model can accurately interpret the prompt and generate semantically consistent results. Our method also surpasses the second-best approach by 1.3 points in the Total Score, demonstrating the effectiveness of the proposed Surprise-driven architecture. In addition, we evaluate long-video generation on VBench-Long [11] by generating 60-second videos, as shown in Tab. 1. At comparable generation speeds, our method achieves significant quality improvements in long video generation, while maintaining the highest subject consistency and aesthetic quality.

Qualitative Analysis. In Fig. 5, we select a challenging example that exhibits continuous motion with non-trivial amplitude over a 1-minute duration to demonstrate the advantage of our method over prior approaches. CausVid suffers from severe error accumulation and memory degradation: the generated video exhibits noticeable color drift and a substantial drop in visual quality over time. Self-Forcing alleviates error accumulation to some extent, yet the visual fidelity still deteriorates markedly once generation exceeds the training horizon, leading to heavy distortions. Rolling Forcing and LongLive largely mitigate error accumulation and significantly improve the visual quality of long-video gener-

ation; however, both suffer from suboptimal quality and inconsistent temporal coherence to varying degrees. For Rolling Forcing, the main subject tends to drift toward the right and frequently exits the frame, which we attribute to limitations of its rolling denoising strategy and scheduling design. LongLive shows insufficient subject consistency, with frequent changes in subject details during long-video generation, likely due to inadequate long-term context management that discards substantial intermediate historical information. In contrast, our method maintains stability and consistency throughout long-video generation, benefiting from a surprise-driven memory management mechanism.

4.3 Ablation Studies

In this section, we analyze the strategies and modules proposed in this work. We first investigate the **Budget-Norm Gating** mechanism by comparing a content-adaptive gating adjustment strategy with a fixed absolute-threshold strategy, in order to examine their impact on the overall generation quality. Next, we study the **Priority-Based Eviction** strategy by comparing it with the commonly used FIFO eviction policy adopted in prior work, and analyze how different eviction mechanisms influence the final generation results, thereby demonstrating the advantage of the proposed method. Finally, we explore the effect of different values of r_{skip} in **Adaptive Step Scheduling** on generation performance. All experiments are conducted on the official VBench-Long (30s) prompt set. More ablation studies and implementation details can be found in the supplementary material.

Table 2: Ablation study of fixed gating and of FIFO and priority-based eviction. **Table 3:** Ablation study of different skip ratios.

Method	Total Score $\uparrow$	Quality Score $\uparrow$	Semantic Score $\uparrow$	Method	Total Score $\uparrow$	Quality Score $\uparrow$	Semantic Score $\uparrow$	Skip Ratio	Total Score $\uparrow$	Quality Score $\uparrow$	Semantic Score $\uparrow$	Acc. Ratio
$z_{thres} = 0.1$	81.24	81.37	80.7	FIFO	82.92	83.44	80.84	$r_{skip} = 0.2$	83.36	83.85	81.40	8.7%
$z_{thres} = 2.0$	81.66	81.83	80.98	Priority	83.20	83.70	81.21	$r_{skip} = 0.4$	83.20	83.70	81.21	15.8%
Budget-Norm	83.20	83.70	81.21					$r_{skip} = 0.6$	82.15	82.46	80.93	21.6%
								$r_{skip} = 0$	84.03	84.55	81.95	0%

Budget-Norm Gating. Our budget-norm gate normalizes surprise scores to z -scores and adjusts the acceptance threshold via closed-loop feedback to maintain a target write rate $r=0.3$ regardless of content dynamics. To validate this design, we compare against two fixed-threshold baselines that retain the same EMA statistics (μ , σ) but replace the adaptive threshold with a static z_{thres} : *Fixed-Low* ($z_{thres} = 0.1$, permissive) and *Fixed-High* ($z_{thres} = 2.0$, selective). All other components remain identical. As shown in Tab. 2, both fixed variants degrade quality. The permissive gate floods the bank with uninformative frames, diluting stored content and reducing routing effectiveness—the bank degenerates toward a uniform sliding window. The selective gate leaves the bank severely underutilized, forfeiting long-range coherence. Our budget-norm mechanism avoids both extremes by automatically converging to the target write rate across diverse

contents, confirming that closed-loop write control is essential: a fixed z -score threshold cannot simultaneously handle high-motion sequences (high surprise) and static scenes (low surprise).

Priority-Based Eviction. When the bank is full, an eviction policy decides which slot to replace. The most common strategy in prior long-context systems is FIFO, which discards the oldest entry regardless of other factors. Our priority-based eviction instead scores each slot by a weighted composite of surprise, usage frequency, and age, replacing the least valuable entry rather than the oldest. We construct a FIFO baseline by zeroing all priority weights except age ($w_a=1.0$), reducing eviction to $p_i = -a_i$. As shown in Tab. 3, FIFO eviction degrades temporal coherence: the bank loses informative early-generation frames (*e.g.*, establishing shots) that are still actively retrieved, while retaining recent but redundant entries that duplicate the sliding-window context. The priority-based policy avoids these failure modes by continuously re-evaluating slot utility, showing that intelligent eviction is essential to selective admission.

Adaptive Step Scheduling. The target skip ratio r_{skip} is the sole hyperparameter of SAD, controlling what fraction of chunks receive a reduced denoising schedule. We sweep $r_{skip} \in \{0.2, 0.4, 0.6\}$ on VBench-Long to characterize the quality-speed tradeoff. As shown in Tab. 4, the speedup scales near-linearly with r_{skip} , reaching 21.6% at $r_{skip} = 0.6$. Even under the most aggressive setting, over 40% of chunks still receive the full 4-step schedule, indicating that the difficulty signal reliably identifies high-motion segments. Considering both acceleration and quality, we adopt $r_{skip} = 0.4$ as the default, which reduces denoising compute by 15.8% with negligible quality degradation.

5 Conclusion

We propose **Surprise Forcing**, a training-free framework for streaming autoregressive video diffusion that decides what to remember and when to skip. By combining a **Surprise-Gated Memory Bank** with **Surprise-Aware Denoising**, our method improves long-horizon consistency while reducing redundant denoising. Future work includes learning more calibrated surprise estimators and extending to multi-level memory and tighter integration with training.

Acknowledgements

This work was supported in part by JSPS KAKENHI Grant Number 24KK0209 and by the Forest Digital Twin Project under the Partnership Agreement for Social Value Creation between UTokyo and SMBC Group.

References

1. Cai, S., Yang, C., Zhang, L., Guo, Y., Xiao, J., Yang, Z., Xu, Y., Yang, Z., Yuille, A., Guibas, L., et al.: Mixture of contexts for long video generation. arXiv preprint arXiv:2508.21058 (2025)

2. Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., Sitzmann, V.: Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems* **37**, 24081–24125 (2024)
3. Chen, G., Lin, D., Yang, J., Lin, C., Zhu, J., Fan, M., Zhang, H., Chen, S., Chen, Z., Ma, C., et al.: Skyreels-v2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074* (2025)
4. Cui, J., Wu, J., Li, M., Yang, T., Li, X., Wang, R., Bai, A., Ban, Y., Hsieh, C.J.: Self-forcing++: Towards minute-scale high-quality video generation. *arXiv preprint arXiv:2510.02283* (2025)
5. Cui, J., Wu, J., Li, M., Yang, T., Li, X., Wang, R., Bai, A., Ban, Y., Hsieh, C.J.: Lol: Longer than longer, scaling video generation to hour. *arXiv preprint arXiv:2601.16914* (2026)
6. Henschel, R., Khachatryan, L., Poghosyan, H., Hayrapetyan, D., Tadevosyan, V., Wang, Z., Navasardyan, S., Shi, H.: Streamingt2v: Consistent, dynamic, and extendable long video generation from text. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 2568–2577 (2025)
7. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *Advances in neural information processing systems* **33**, 6840–6851 (2020)
8. Huang, J., Hu, X., Han, B., Shi, S., Tian, Z., He, T., Jiang, L.: Memory forcing: Spatio-temporal memory for consistent scene generation on minecraft. *arXiv preprint arXiv:2510.03198* (2025)
9. Huang, X., Li, Z., He, G., Zhou, M., Shechtman, E.: Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009* (2025)
10. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., et al.: Vbench: Comprehensive benchmark suite for video generative models. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 21807–21818 (2024)
11. Huang, Z., Zhang, F., Xu, X., He, Y., Yu, J., Dong, Z., Ma, Q., Chanpaisit, N., Si, C., Jiang, Y., Wang, Y., Chen, X., Chen, Y.C., Wang, L., Lin, D., Qiao, Y., Liu, Z.: VBench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025)
12. Ji, S., Chen, X., Yang, S., Tao, X., Wan, P., Zhao, H.: Memflow: Flowing adaptive memory for consistent and efficient long video narratives. *arXiv preprint arXiv:2512.14699* (2025)
13. Jiang, J., Li, W., Ren, J., Qiu, Y., Pei, R., Song, F., Guo, Y., Xu, X., Wu, H., Zuo, W.: Lovic: Efficient long video generation with context compression. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 4022–4034 (2026)
14. Kahatapitiya, K., Liu, H., He, S., Liu, D., Jia, M., Zhang, C., Ryoo, M.S., Xie, T.: Adaptive caching for faster video generation with diffusion transformers. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 15240–15252 (2025)
15. Kim, J., Kang, J., Choi, J., Han, B.: Fifo-diffusion: Generating infinite videos from text without training. *Advances in Neural Information Processing Systems* **37**, 89834–89868 (2024)
16. Kodaira, A., Hou, T., Hou, J., Georgopoulos, M., Juefei-Xu, F., Tomizuka, M., Zhao, Y.: Streamdit: Real-time streaming text-to-video generation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 29200–29210 (2026)

17. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.t., Rocktäschel, T., et al.: Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* **33**, 9459–9474 (2020)
18. Li, R., Torr, P., Vedaldi, A., Jakab, T.: Vmem: Consistent interactive video scene generation with surfel-indexed view memory. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 25690–25699 (2025)
19. Li, X., Li, M., Cai, T., Xi, H., Yang, S., Lin, Y., Zhang, L., Yang, S., Hu, J., Peng, K., et al.: Radial attention: $\mathcal{O}(n \log n)$ sparse attention with energy decay for long video generation. *Advances in Neural Information Processing Systems* **38**, 16822–16852 (2026)
20. Liu, K., Hu, W., Xu, J., Shan, Y., Lu, S.: Rolling forcing: Autoregressive long video diffusion in real time. *arXiv preprint arXiv:2509.25161* (2025)
21. Lu, Y., Liang, Y., Zhu, L., Yang, Y.: Freelong: Training-free long video generation with spectralblend temporal attention. *Advances in Neural Information Processing Systems* **37**, 131434–131455 (2024)
22. Lu, Y., Yang, Y.: Freelong++: Training-free long video generation via multi-band spectralfusion. *arXiv preprint arXiv:2507.00162* (2025)
23. Polyak, A., Zohar, A., Brown, A., Tjandra, A., Sinha, A., Lee, A., Vyas, A., Shi, B., Ma, C.Y., Chuang, C.Y., et al.: Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720* (2024)
24. Qiu, H., Liu, S., Zhou, Z., An, Z., Ren, W., Liu, Z., Schult, J., He, S., Chen, S., Cong, Y., et al.: Histream: Efficient high-resolution video generation via redundancy eliminated streaming. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 4603–4613 (2026)
25. Qiu, H., Xia, M., Zhang, Y., He, Y., Wang, X., Shan, Y., Liu, Z.: Freenoise: Tuning-free longer video diffusion via noise rescheduling. In: *International Conference on Learning Representations*. vol. 2024, pp. 5260–5274 (2024)
26. Ruhe, D., Heek, J., Salimans, T., Hooeboom, E.: Rolling diffusion models. *arXiv preprint arXiv:2402.09470* (2024)
27. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020)
28. Song, K., Chen, B., Simchowitz, M., Du, Y., Tedrake, R., Sitzmann, V.: History-guided video diffusion. *arXiv preprint arXiv:2502.06764* (2025)
29. Tan, S., Gong, B., Feng, Y., Zheng, K., Zheng, D., Shi, S., Shen, Y., Chen, J., Yang, M.: Mimir: Improving video diffusion models for precise text understanding. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 23978–23988 (2025)
30. Teng, H., Jia, H., Sun, L., Li, L., Li, M., Tang, M., Han, S., Zhang, T., Zhang, W., Luo, W., et al.: Magi-1: Autoregressive video generation at scale. *arXiv preprint arXiv:2505.13211* (2025)
31. Wan, T., Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., et al.: Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314* (2025)
32. Wu, B., Zou, C., Li, C., Huang, D., Yang, F., Tan, H., Peng, J., Wu, J., Xiong, J., Jiang, J., et al.: Hunyuanvideo 1.5 technical report. *arXiv preprint arXiv:2511.18870* (2025)
33. Wu, X., Zhang, G., Xu, Z., Zhou, Y., Lu, Q., He, X.: Pack and force your memory: Long-form and consistent video generation. *arXiv preprint arXiv:2510.01784* (2025)

34. Xi, H., Yang, S., Zhao, Y., Xu, C., Li, M., Li, X., Lin, Y., Cai, H., Zhang, J., Li, D., et al.: Sparse videogen: Accelerating video diffusion transformers with spatial-temporal sparsity. arXiv preprint arXiv:2502.01776 (2025)
35. Xiao, G., Tian, Y., Chen, B., Han, S., Lewis, M.: Efficient streaming language models with attention sinks. In: International Conference on Learning Representations. vol. 2024, pp. 21875–21895 (2024)
36. Xiao, Z., Lan, Y., Zhou, Y., Ouyang, W., Yang, S., Zeng, Y., Pan, X.: Worldmem: Long-term consistent world simulation with memory. Advances in Neural Information Processing Systems **38**, 49632–49652 (2026)
37. Yang, S., Huang, W., Chu, R., Xiao, Y., Zhao, Y., Wang, X., Li, M., Xie, E., Chen, Y., Lu, Y., et al.: Longlive: Real-time interactive long video generation. arXiv preprint arXiv:2509.22622 (2025)
38. Yang, Z., Teng, J., Zheng, W., Ding, M., Huang, S., Xu, J., Yang, Y., Hong, W., Zhang, X., Feng, G., et al.: Cogvideox: Text-to-video diffusion models with an expert transformer. arXiv preprint arXiv:2408.06072 (2024)
39. Yi, J., Jang, W., Cho, P.H., Nam, J., Yoon, H., Kim, S.: Deep forcing: Training-free long video generation with deep sink and participative compression. arXiv preprint arXiv:2512.05081 (2025)
40. Yin, T., Gharbi, M., Park, T., Zhang, R., Shechtman, E., Durand, F., Freeman, B.: Improved distribution matching distillation for fast image synthesis. Advances in neural information processing systems **37**, 47455–47487 (2024)
41. Yin, T., Gharbi, M., Zhang, R., Shechtman, E., Durand, F., Freeman, W.T., Park, T.: One-step diffusion with distribution matching distillation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6613–6623 (2024)
42. Yin, T., Zhang, Q., Zhang, R., Freeman, W.T., Durand, F., Shechtman, E., Huang, X.: From slow bidirectional to fast autoregressive video diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22963–22974 (2025)
43. Yu, J., Bai, J., Qin, Y., Liu, Q., Wang, X., Wan, P., Zhang, D., Liu, X.: Context as memory: Scene-consistent interactive long video generation with memory retrieval. In: Proceedings of the SIGGRAPH Asia 2025 Conference Papers. pp. 1–11 (2025)
44. Zhang, L., Cai, S., Li, M., Wetzstein, G., Agrawala, M.: Frame context packing and drift prevention in next-frame-prediction video diffusion models. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025)
45. Zhang, L., Cai, S., Li, M., Zeng, C., Lu, B., Rao, A., Han, S., Wetzstein, G., Agrawala, M.: Pretraining frame preservation in autoregressive video memory compression. arXiv preprint arXiv:2512.23851 (2025)