SELF-ALIGNED FORCING: STREAMING VIDEO DIFFUSION WITH DIFFERENTIABLE NOISY HISTORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Autoregressive video diffusion enables interactive streaming generation, but suffers from error accumulation over long rollouts. Self-rollout training reduces exposure bias, yet finite rollouts leave long-range drift unresolved. We observe that the noise level of the history key-value (K/V) representations trades visual quality against motion, and that restoring gradients through the history aligns causal training far more closely with bidirectional training. Motivated by these observations, we introduce **Self-Aligned Forcing** (**SAF**), a training scheme that *aligns* the history of each block with the noise level of the block being denoised. Specifically, the history is the K/V produced by preceding blocks at the same denoising stage, so all blocks at a stage can be denoised in a single forward pass under a causal mask. This keeps the noisy history differentiable, allowing future losses to optimize how it is encoded. SAF therefore avoids a separate no-gradient rollout and per-block timestep-zero recaching, training up to $1.8\times$ faster than prior methods with lower memory. At inference, SAF achieves the highest single-GPU throughput among existing methods and keeps one history bank per stage for a multi-GPU pipeline, reaching 49.1 FPS on 4 GPUs. Experiments show superior long-horizon generation with a better balance between visual quality and motion.

1 INTRODUCTION

Autoregressive (AR) video diffusion has emerged as a promising approach to interactive streaming video generation (Yin et al., 2025; Huang et al., 2025; Yang et al., 2026). It generates a video block by block, conditioning each new block on the key-value (K/V) representations of previously generated blocks, which form the *history*. The model can thus continuously extend a video and respond to new instructions without reprocessing the full sequence, as required by interactive applications such as game engines and world models (Bruce et al., 2024; Valevski et al., 2025; He et al., 2025). However, long-horizon streaming remains challenging, as it requires stability over videos much longer than the training clips, meaningful motion, and efficient generation.

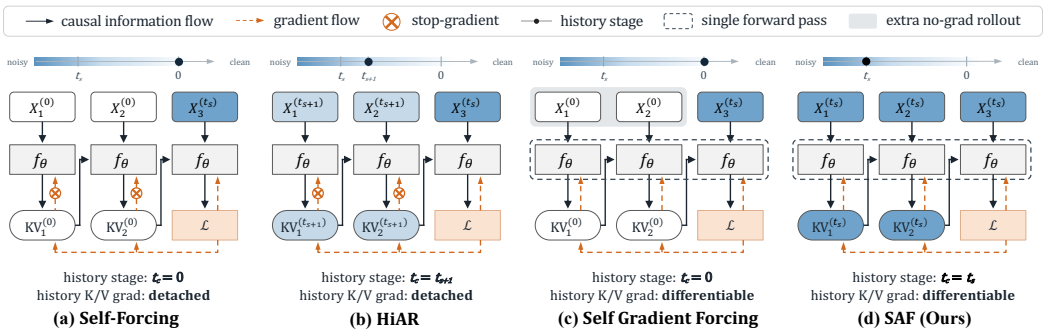
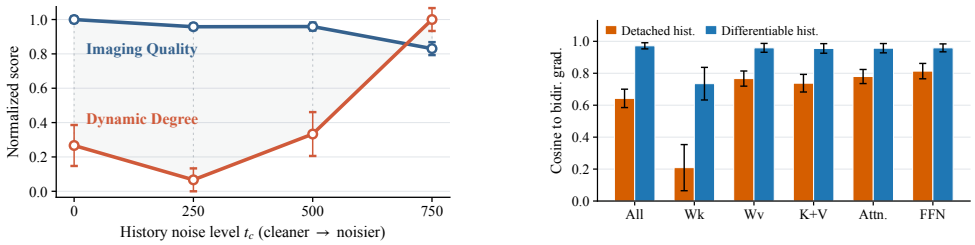


Figure 1: **History stage and gradient flow.** The model f_θ generates blocks from left to right, and the K/V of earlier blocks form the history of the last block, whose loss $\mathcal{L}$ is shown. Superscripts give noise levels (t_s: current block, t_c: history, 0: clean). Self-Forcing **(a)** and HiAR **(b)** detach the history, so the loss cannot reach it; Self Gradient Forcing **(c)** and SAF **(d)** keep this gradient with the history at $t_c = 0$ and $t_c = t_s$, respectively, but the former needs an extra no-grad rollout.

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.



(a) Noisier history adds motion but lowers quality. (b) History gradients align with bidirectional training.

Figure 2: **Two observations motivating differentiable noisy history. (a)** A fixed LongLive model with varying history noise level t_c; imaging quality and dynamic degree (Huang et al., 2024) are normalized to their values at $t_c = 0$ and $t_c = 750$, respectively. **(b)** With all settings identical except whether gradients flow through the history K/V, differentiable history raises the gradient cosine similarity to a matched-weight bidirectional reference from 0.643 to 0.972. Error bars show the range over multiple samples.

Long-horizon stability is limited mainly by error accumulation. Self-Forcing (Huang et al., 2025) narrows the train–test gap, known as exposure bias (Ranzato et al., 2016), by training the model on its own generated history (Figure 1a). Nevertheless, its finite rollouts do not fully account for errors accumulated over longer horizons. To improve robustness under limited training rollouts, recent methods build the history from partially denoised states. Rolling Forcing (Liu et al., 2026) jointly denoises a rolling window with staggered noise levels, while HiAR (Zou et al., 2026) takes the history one denoising step cleaner than the current block (Figure 1b). HiAR further finds that aligning the history noise level with that of the current block minimizes error accumulation but lowers the overall VBench quality score (Huang et al., 2024).

Observation 1. Noisy history not only reduces error accumulation but also increases motion dynamics, at the cost of imaging quality. Using a LongLive checkpoint trained with clean history, we vary only the history noise level t_c at inference. As shown in Figure 2a, sufficiently noisy history substantially increases the VBench (Huang et al., 2024) dynamic degree, but imaging quality drops. The motion gain is notable, since models trained with distribution matching distillation (DMD) (Yin et al., 2024a;b) commonly produce static outputs due to its mode-seeking objective (Zou et al., 2026; You et al., 2026; Wu et al., 2026). HiAR attributes reduced error accumulation to the lower signal-to-noise ratio (SNR) of noisy history (Zou et al., 2026); our observation suggests that this mechanism also underlies the motion gain and the quality loss. By weakening the constraint that the history imposes on the current block, lower SNR suppresses error propagation and increases motion dynamics, but also reduces temporal consistency and lowers imaging quality. The benefit and the cost thus share a single cause, yet the model cannot learn to balance them.

Observation 2. Detaching the history prevents the model from learning how to encode it for future blocks. Existing self-rollout methods detach the history K/V from gradient computation (Figure 1a,b), as backpropagating through it would nest the computation graphs of all preceding blocks. Future losses therefore cannot optimize how preceding blocks are encoded for subsequent prediction. To quantify this effect, we compare the resulting parameter gradient with a matched-weight bidirectional reference, in which all blocks are processed jointly so that losses on later blocks also supervise earlier ones. With all other settings identical, restoring the gradient path through the history increases their cosine similarity from 0.643 to 0.972 (Figure 2b). The gap is largest for the key projection W_k, whose cosine similarity drops to about 0.2 under detachment, indicating that the model barely learns to encode the history for future retrieval. Noisy history should therefore remain differentiable rather than serve only as fixed conditioning.

Existing methods face a dilemma: noisy history is robust but detached, whereas differentiable history requires costly reconstruction. Self Gradient Forcing (SGF) (Zhuang et al., 2026) restores gradients through the history (Figure 1c), but requires a separate no-gradient rollout to construct the history (similar to SDF in Figure 7). Moreover, its history is built from clean states, available only after a block is fully denoised and recached at timestep zero. This recaching adds a forward pass per block and prevents pipelining across denoising stages, limiting both training and inference efficiency.

Our key insight is that stage alignment between the history and the current block makes noisy history differentiable efficiently without reconstruction. The history K/V is then exactly what preceding

blocks compute while being denoised at the same stage. Since the history and the current block share one noise level, they can be processed in a single forward pass (Figures 1d and 3b), so the history is both noisy and differentiable without reconstruction. Besides, this causal training strategy shares the structure of bidirectional diffusion training, in which all blocks share one noise level and later blocks supervise earlier ones. Notably, this aligned setting lowers quality in HiAR, where the history is detached; once the history is learned, it achieves the highest Quality under every inference policy among the history schedules we study in training (Section 4.3).

Building on this insight, we propose **Self-Aligned Forcing** (SAF). SAF rolls out all blocks stage by stage: at each denoising step, a single block-causal forward pass processes every block, and gradients are taken at one randomly sampled stage under the standard DMD objective. Apart from a clean attention sink (Xiao et al., 2024) recached within the same forward pass, it needs neither per-block recaching nor a separate no-gradient rollout, reducing training time and memory (Table 3). At inference, SAF can keep one history bank per noise level for a multi-GPU pipeline, or a single bank on one GPU to save memory (Figure 3c). Our contributions are threefold:

- We study how the history should be represented and reused in autoregressive video diffusion, showing that propagating gradients through noisy history K/V brings causal training close to bidirectional training, improving both overall quality and motion dynamics.
- We propose **SAF**, which aligns the history noise level with the current block. All blocks are then rolled out in parallel within each stage, and gradients at a sampled stage can flow through the history. A clean sink anchors appearance at negligible cost, and per-stage history banks enable pipelined inference across GPUs.
- We evaluate SAF extensively on multiple benchmarks. It performs best in the chunkwise setting and competitively in the framewise setting, achieving the highest training and inference efficiency in both settings, e.g., $1.8\times$ faster framewise training than SGF and 49.1 FPS in chunkwise four-GPU pipelined inference.

2 Related Work

Autoregressive Video Diffusion. Autoregressive video diffusion generates a video as causal temporal blocks, enabling streaming generation with a KV-cached history (Sand AI et al., 2025; Lin et al., 2025; Kodaira et al., 2026; Lu et al., 2026; Yi et al., 2026). Diffusion Forcing (Chen et al., 2024) assigns different noise levels to temporal tokens, CausVid (Yin et al., 2025) distills a bidirectional model into a few-step causal generator, and LongLive (Yang et al., 2026) scales this paradigm to real-time long videos with streaming long tuning, windowed attention, frame sinks, and KV recaching. To improve long-horizon robustness, Rolling Forcing (Liu et al., 2026) jointly denoises a window at staggered noise levels, and HiAR (Zou et al., 2026) denoises timestep-first and builds the history from intermediate denoising states. These methods change the noise level of the history but keep it detached, so losses on later blocks cannot optimize how it is encoded.

Self-Generated Rollout and Cross-Block Credit Assignment. Self-Forcing (Huang et al., 2025) trains the model on its own generated history to narrow the train–test gap, and Resampling Forcing (Guo et al., 2026) and BAgger (Po et al., 2026) further improve robustness through self-resampling and corrective rollout trajectories. All of them, however, detach the history K/V, so future losses cannot shape how earlier blocks are encoded. Live Avatar (Huang et al., 2026a) also matches the history timestep to the current block, yet its rollout remains serial and its history detached. Video-Mirai (Yu et al., 2026) adds future-aware supervision through an auxiliary foresight pathway and feature distillation, whereas SGF (Zhuang et al., 2026) restores future-to-history gradients by re-encoding clean history built in a separate no-gradient rollout. Existing methods thus either detach the history, add auxiliary supervision, or require a separate no-gradient rollout, leaving efficient future-to-history optimization of noisy history K/V unresolved.

3 Method

3.1 Overview

SAF rests on a single design choice: at every denoising step, the history is taken at the same noise level as the current block. This choice makes the local history both noisy and differentiable without

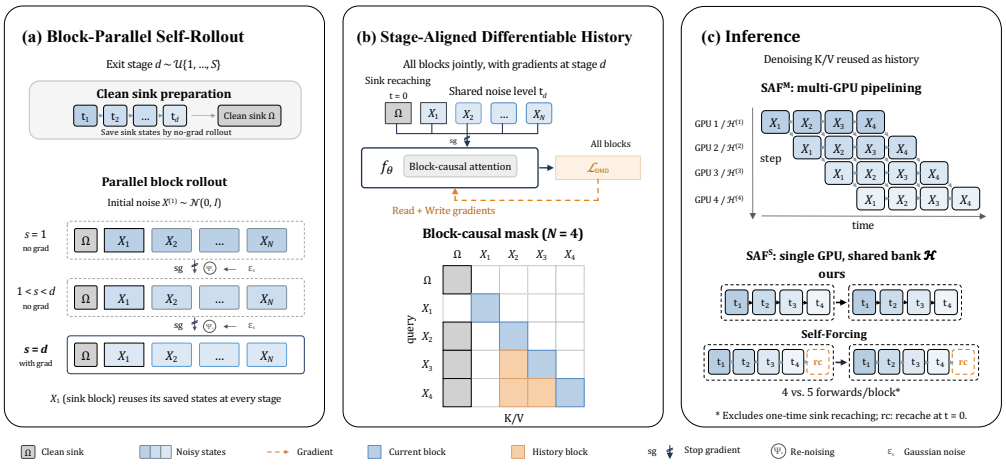


Figure 3: **Overview of SAF. (a)** Rollout is parallel across blocks and sequential only across stages (Section 3.3). **(b)** Stage alignment enables joint denoising with differentiable history in one block-causal forward pass (Section 3.2). **(c)** Reusing the K/V computed during denoising as history allows multi-GPU stage pipelining and removes per-block recaching (Section 3.5).

any reconstruction (Section 3.2) and allows self-rollout to proceed in parallel across blocks during training (Section 3.3). We complement it with a clean sink that anchors appearance over the whole video (Section 3.4), and at inference it yields per-stage history K/V banks that pipeline naturally across GPUs (Section 3.5). Figure 3 summarizes the framework.

A video is generated as N blocks, each denoised at noise levels $t_1 > \cdots > t_S$, where $t_1 = 1000$ is pure noise; we call the step at t_s stage s. At stage s, the causal denoiser f_θ reads the noisy latent $X_i^{(s)}$ of block i, its history K/V bank $\mathcal{H}_i^{(s)}$, and the text prompt p, predicts a clean-latent estimate $\widehat{X}_i^{(s)}$, and produces the block's own $\mathrm{KV}_i^{(s)}$. We write this step as $(\widehat{X}_i^{(s)}, \mathrm{KV}_i^{(s)}) = f_\theta(X_i^{(s)}, t_s; \mathcal{H}_i^{(s)}, p)$. $\mathcal{H}_i^{(s)}$ holds the KV_Ω of a persistent sink Ω (the first few blocks (Xiao et al., 2024)) and the K/V of the latest preceding blocks, which together with block i span a local window of L frames.

3.2 Stage-Aligned Differentiable History

Observations 1 and 2 call for a history that is noisy yet differentiable, but serial self-rollout makes these requirements conflict. When blocks are generated one after another, the history of block i is produced by earlier forward passes, and keeping it differentiable would nest the computation graphs of all preceding blocks. Existing methods therefore either detach a noisy history (Zou et al., 2026) or construct a clean one in a separate no-gradient rollout (Zhuang et al., 2026).

Stage alignment resolves this conflict by letting the history be produced and consumed in the same forward pass (Figures 1d and 3b). When block i reads its history at its own noise level t_s, the K/V of each predecessor j is exactly what block j computes for its own prediction at stage s. A single block-causal forward pass over all blocks therefore yields every prediction together with every history:

$$\big(\widehat{X}^{(s)},\, \mathrm{KV}^{(s)}\big) = f_\theta\big(\mathrm{sg}(X^{(s)}), t_s; M_{\mathrm{causal}}, p\big), \tag{1}$$

where $X^{(s)}$, $\widehat{X}^{(s)}$, and $\mathrm{KV}^{(s)}$ stack the inputs, estimates, and K/V of all blocks and sg is stop-gradient. The block-causal mask M_{causal} lets each block i attend only to itself and its history $\mathcal{H}_i^{(s)} = \{\mathrm{KV}_j^{(s)}\}_{j \in \mathcal{W}(i)}$, the same-stage K/V of its predecessors within the window $\mathcal{W}(i)$. Eq. 1 is thus the per-block step of Section 3.1 applied to all blocks at once, and within a stage, training resembles bidirectional diffusion training with only the future masked out. For clarity, we omit the sink Ω here and describe it in Section 3.4.

Because Eq. 1 computes the history rather than loading it from a detached cache, the gradient of $\widehat{X}_i^{(s)}$ with respect to the parameters θ splits into two terms:

$$\frac{\mathrm{d}\widehat{X}_i^{(s)}}{\mathrm{d}\theta} = \underbrace{\frac{\partial \widehat{X}_i^{(s)}}{\partial \theta}}_{\text{read}} + \underbrace{\sum_{h \in \mathcal{H}_i^{(s)}} \frac{\partial \widehat{X}_i^{(s)}}{\partial h} \frac{\mathrm{d}h}{\mathrm{d}\theta}}_{\text{write}}, \tag{2}$$

where h ranges over the K/V entries of $\mathcal{H}_i^{(s)}$. Viewing the history as a memory that predecessors write and block i reads, the read term updates how block i uses the history, while the write term updates how its predecessors encode it, providing future-to-history supervision. As the inputs are detached, the write term changes only how the history is encoded, not what is sampled. The model thus learns to write noisy history rather than merely tolerate it.

3.3 Block-Parallel Self-Rollout

Although each stage of SAF mirrors bidirectional training under a causal mask, the two still differ in the origin of their inputs. Bidirectional training noises ground-truth videos, which carries over to inference because all frames are denoised jointly. A causal model, however, reads a history derived from its own earlier outputs at inference, so training on ground-truth inputs creates a train–test mismatch that lets errors accumulate across blocks (Huang et al., 2025). SAF therefore retains self-rollout: starting from $X^{(1)} \sim \mathcal{N}(0, I)$, the input to each stage is derived from the model's estimate at the preceding stage,

$$X^{(s+1)} = \Psi_s\big(\mathrm{sg}(\widehat{X}^{(s)}), \epsilon_s\big), \tag{3}$$

where the sampler Ψ_s re-noises the estimate with noise level t_{s+1} using Gaussian noise ϵ_s, and $\widehat{X}^{(s)}$ is given by Eq. 1. In contrast to existing serial self-rollout (Huang et al., 2025; Zou et al., 2026), which advances one block per forward pass, SAF parallelizes the rollout across blocks within each stage and keeps it sequential only across stages (Figure 3a).

To avoid storing the computation graphs of all S chained stages, we follow the stochastic gradient truncation of Self-Forcing (Huang et al., 2025): each generator update samples an exit stage $d \sim \mathcal{U}\{1, \ldots, S\}$, covering every noise level in expectation with a single stage in the computation graph. Stages $s < d$ run without gradients, and only stage d evaluates Eq. 1 with gradients under the standard DMD objective (Yin et al., 2024a):

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{DMD}}\big(\widehat{X}^{(d)}, p\big), \tag{4}$$

with score models updated as in DMD (Algorithm 1).

Block-parallel rollout thus keeps the benefits of self-rollout without its serial cost: apart from a short sink pre-roll (Section 3.4), the sequential depth drops from $N \times d$ block-level to d stage-level forward passes, and since the supervised pass itself yields the differentiable history, no separate rollout is needed to build it. Training is therefore faster and more memory-efficient (Table 3), with a cost nearly independent of block granularity since the sequential depth no longer grows with N.

3.4 Clean Sink as a Persistent Anchor

Noisy history suits the local window, which is refreshed at every block and only needs to convey how the scene evolves. The sink Ω, in contrast, is the only memory that persists beyond the window; it alone anchors subject identity and scene layout and thus calls for a high signal-to-noise ratio (SNR). We therefore give every stage a clean sink.

Since the rollout stops at the exit stage d (Section 3.3), the cleanest available sink is its own estimate $\widehat{X}_\Omega^{(d)}$. Because every stage needs this estimate, we first roll out the sink blocks sequentially to stage d without gradients, recording their inputs $X_\Omega^{(s)}$ for $s \le d$ (Figure 3a). Each stage then prepends the clean sink at timestep zero to its input:

$$Z^{(s)} = \big[\ \underbrace{\mathrm{sg}(\widehat{X}_\Omega^{(d)})}_{\text{clean sink, } t=0}\ ;\ \underbrace{\mathrm{sg}(X^{(s)})}_{\text{all blocks, } t=t_s}\ \big], \tag{5}$$

where the sink blocks inside $X^{(s)}$ reuse the recorded $X_\Omega^{(s)}$, so each sink block appears both clean and at noise level t_s. Eq. 1 then takes $Z^{(s)}$ in place of $\mathrm{sg}(X^{(s)})$, and the same forward pass re-encodes the clean sink into KV_Ω instead of reading a detached cache. We call this *sink recaching*. All stages thus share the same clean sink, whose K/V stays in the computation graph at the exit stage. The history of Section 3.2 thus becomes $\mathcal{H}_i^{(s)} = \mathrm{KV}_{\Omega,<i} \cup \{\mathrm{KV}_j^{(s)}\}_{j\in\mathcal{W}(i)}$, where $\mathrm{KV}_{\Omega,<i}$ is the clean K/V of the sink blocks preceding block i and $\mathcal{W}(i)$ now covers only non-sink predecessors. For a sink block, $\mathcal{W}(i)$ is empty, so the first sink block has no history.

The sink therefore plays two roles. As content, its noisy copy is predicted like any other block and receives the DMD loss; as memory, its clean copy supplies KV_Ω to later blocks and receives the write term of Eq. 2. At inference, only the sink is recached once per video from its final estimate $\widehat{X}_\Omega^{(S)}$, while clean-history recaching (Huang et al., 2025; Zhuang et al., 2026) re-encodes every block at timestep zero. This clean anchor stabilizes appearance without suppressing motion, whereas a noisy sink collapses generation to nearly static videos (Section 4.3).

3.5 Streaming Inference

Multi-bank streaming. To keep latency low, SAF generates the video in a streaming manner at inference, emitting each block once it has passed all S stages instead of running Eq. 1 over the whole sequence. Nevertheless, each block should read the history seen in training, namely one at its current noise level. Multi-bank SAF (SAF^{M}) therefore maintains one history K/V bank $\mathcal{H}^{(s)}$ per stage (Figure 3c), which holds exactly $\mathcal{H}_i^{(s)}$ when block i reaches stage s:

$$\big(\widehat{X}_i^{(s)}, \mathrm{KV}_i^{(s)}\big) = f_\theta\big(X_i^{(s)}, t_s; \mathcal{H}^{(s)}, p_i\big), \qquad \mathcal{H}^{(s)} \leftarrow \mathrm{Commit}_L\big(\mathcal{H}^{(s)}, \mathrm{KV}_i^{(s)}\big), \tag{6}$$

where p_i is the prompt of block i, and Commit_L appends the new K/V after the forward pass and evicts the oldest non-sink entries beyond the window. All banks share the same clean sink. Because the K/V computed during denoising is reused as history, each block needs only S forward passes rather than the $S{+}1$ required by clean-history recaching (Huang et al., 2025; Zhuang et al., 2026).

Multi-GPU stage pipelining. Since block i at stage s depends only on its latent from stage $s{-}1$ and $\mathcal{H}^{(s)}$, stages decouple across GPUs: GPU s holds a copy of f_θ and $\mathcal{H}^{(s)}$, denoises blocks in order at stage s, and passes each latent to GPU $s{+}1$ while keeping the K/V locally (Figure 3c and Algorithm 2). While GPU s processes block i, GPU $s{+}1$ processes block $i{-}1$. Once the pipeline is full, S blocks are in flight, each at a different stage, and each GPU stores one history K/V bank. Clean-history methods (Huang et al., 2025; Zhuang et al., 2026) cannot be pipelined this way, since block $i{+}1$ must wait until block i completes all stages and is recached at $t = 0$.

Single-bank streaming. When memory is limited, single-bank SAF (SAF^{S}) runs the same model on one GPU with a single bank $\mathcal{H}$ shared by all stages (Figure 3c and Algorithm 4). Once a block is fully denoised, only the K/V from one of its stages is committed to $\mathcal{H}$, with that stage's noise level either fixed to t (Fixed$t$) or sampled per block. Since noisier history K/V favors dynamics and cleaner history K/V favors fidelity (Figure 2a), SAF^{S} balances the two by sampling noise levels $(250, 500, 750)$ with probabilities $(0.5, 0.25, 0.25)$, which we call Mix. This variant saves memory but cannot be pipelined, and most stages read history at a noise level mismatched with training, so SAF^{M} remains our default.

4 Experiments

4.1 Experimental Setup

Our models are trained on 5-second, single-prompt video clips in both framewise and chunkwise settings, and we evaluate whether this short-horizon training transfers to longer and interactive generation. Specifically, three complementary benchmarks are used: **VBench** (Huang et al., 2024) evaluates 5-second clips at the training horizon, **Interactive** (Yang et al., 2026; Ji et al., 2025) generates 60-second videos across six 10-second segments, each conditioned on a distinct prompt, and **MovieGen-100s** (Polyak et al., 2025; Zhang et al., 2026) generates 100-second videos from 128 prompts to test generation quality at $20\times$ the training length. We compare with Self-Forcing (Huang et al., 2025), LongLive (Yang et al., 2026), Causal-Forcing (Zhu et al., 2026), HiAR (Zou et al.,

Algorithm 1: Parallel Training of SAF

Require: Denoiser f_θ, stages $t_{1:S}$, sampler Ψ, N blocks, causal mask M_{causal}, sink Ω, prompt p
Ensure: One training iteration

Sample exit stage $d \sim \mathcal{U}\{1, \dots, S\}$
Sample $X^{(1)}, \epsilon_{1:d-1} \sim \mathcal{N}(0, I)$
for sink block $i \in \Omega$ in order **do** ▷ no gradients
  for $s = 1, \dots, d$ **do**
    $\widehat{X}_i^{(s)} \leftarrow f_\theta(X_i^{(s)}, t_s; \mathcal{H}_i^{(s)}, p)$
    $X_i^{(s+1)} \leftarrow \Psi_s(\widehat{X}_i^{(s)}, \epsilon_s)$ if $s < d$
  end for
end for
for $s = 1, \dots, d$ **do**
  $Z^{(s)} \leftarrow [\mathrm{sg}(\widehat{X}_\Omega^{(d)}); \mathrm{sg}(X^{(s)})]$
  clean sink at $t = 0$, all blocks at t_s
  if $s < d$ **then** ▷ no gradients
    $\widehat{X}^{(s)} \leftarrow f_\theta(Z^{(s)}, t_s; M_{\text{causal}}, p)$
    $X^{(s+1)} \leftarrow \Psi_s(\widehat{X}^{(s)}, \epsilon_s)$
    with sink entries reset to $X_\Omega^{(s+1)}$
  else ▷ with gradients
    $\widehat{X}^{(d)} \leftarrow f_\theta(Z^{(d)}, t_d; M_{\text{causal}}, p)$
    sink and history K/V remain differentiable
  end if
end for
$\mathcal{L} \leftarrow \mathcal{L}_{\text{DMD}}(\widehat{X}^{(d)}, p)$
Backpropagate the read and write terms; update θ
Update the score models as in DMD

Algorithm 2: Multi-GPU Pipeline Inference of SAF^{M}

Require: Denoiser f_θ, stages $t_{1:S}$, sampler Ψ, N blocks, window L, sink Ω, block prompts $p_{1:N}$, S GPUs
Ensure: Ordered clean blocks $\widehat{X}_{1:N}^{(S)}$

Disable gradients and replicate f_θ across GPUs
Generate and emit the sink blocks sequentially; encode $\widehat{X}_\Omega^{(S)}$ once at $t = 0$ to obtain KV_Ω
Each GPU s initializes $\mathcal{H}^{(s)} \leftarrow \mathrm{Init}(\mathrm{KV}_\Omega)$
for clock $r = 1, \dots, N - |\Omega| + S - 1$ **do**
  for all GPUs s in parallel **do**
    $i \leftarrow |\Omega| + r - s + 1$; skip if $i \notin (|\Omega|, N]$
    if $s = 1$ **then**
      Sample $X_i^{(1)} \sim \mathcal{N}(0, I)$
    else
      Receive $X_i^{(s)}$ from GPU $s - 1$
    end if
    $(\widehat{X}_i^{(s)}, \mathrm{KV}_i^{(s)}) \leftarrow f_\theta(X_i^{(s)}, t_s; \mathcal{H}^{(s)}, p_i)$
    $\mathcal{H}^{(s)} \leftarrow \mathrm{Commit}_L(\mathcal{H}^{(s)}, \mathrm{KV}_i^{(s)})$
    if $s < S$ **then**
      Sample $\epsilon_{i,s} \sim \mathcal{N}(0, I)$
      $X_i^{(s+1)} \leftarrow \Psi_s(\widehat{X}_i^{(s)}, \epsilon_{i,s})$
      Send $X_i^{(s+1)}$ to GPU $s + 1$
    else
      Emit $\widehat{X}_i^{(S)}$
    end if
  end for
end for

Table 1: **Chunkwise evaluation with existing methods.** FPS is measured on H100 GPUs; when two throughput values are reported, the second denotes 4-GPU pipelined parallel inference. Best results are **bold** and second-best results are underlined.

Method	Throughput (FPS) ↑	VBench			Interactive		MovieGen 100s
		Total ↑	Quality ↑	Semantic ↑	ViCLIP ↑	Quality ↑	Quality ↑
Self-Forcing	17.0	83.80	84.59	80.64	21.88	80.69	78.74
LongLive	20.7	83.22	83.68	**81.37**	**25.74**	82.17	82.83
Causal-Forcing	17.0	84.88	85.93	80.67	24.21	81.66	79.97
HiAR	11.2 / 30.0	82.66	83.34	79.93	24.16	80.97	81.84
SGF	20.7	84.76	85.74	80.83	24.63	83.83	84.63
SAF^{S}	**22.9**	84.61	85.80	79.87	24.70	84.29	84.34
SAF^{M}	22.8 / **49.1**	**85.04**	**86.30**	80.00	25.34	**85.59**	**85.00**

2026), and SGF (Zhuang et al., 2026), re-evaluating all of them with official weights under identical random seeds; only Causal-Forcing and SGF release framewise checkpoints. Additional protocol and scoring details are provided in Appendix C.

4.2 Comparison with Existing Methods

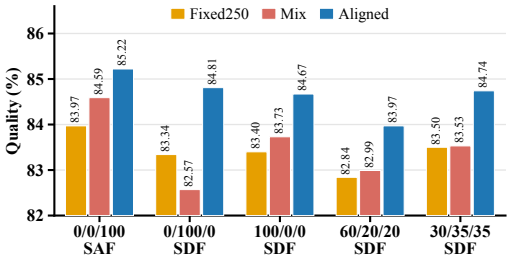
Quantitative analysis. Table 1 reports the chunkwise comparison, where ours with SAF^{M} leads nearly all aggregate metrics across short-video, interactive, and long-video generation. LongLive attains a higher Interactive ViCLIP score, plausibly reflecting its dedicated prompt-switching and long-video training, but ours still achieves higher Interactive Quality and MovieGen-100s Quality. The only remaining deficit is a small drop in VBench Semantic, a metric that often decreases when motion and dynamics are stronger. Ours is also the fastest single-GPU method because it avoids the extra timestep-zero forward pass required by clean-history recaching, and its stage-wise design enables four-GPU pipelined inference for a further throughput gain. SAF^{S} is slightly weaker than SAF^{M} due to its train–test mismatch, yet it remains competitive and preserves clear quality advantages. Full dimension-wise scores and the framewise comparison are provided in Appendix D.

Qualitative analysis. Figures 5 and 6 show that the chunkwise gains translate into stronger semantic alignment and visual stability. In Interactive generation, ours tracks successive prompt changes

Table 2: **Component and K/V bank ablations on 20-second Interactive generation.** At inference, SAF^{M} uses stage-aligned multiple K/V banks and SAF^{S} uses one bank written at noise levels sampled from $(250, 500, 750)$ with probabilities $(0.5, 0.25, 0.25)$. Indented w/o rows: training ablations under the same inference; Fixed$t$: SAF with one bank written at noise level t. Best per group in **bold**, second underlined.

Method	ViCLIP ↑	Quality ↑	Smooth ↑	Dyn. ↑	Img. ↑
SAF^{M}	**26.54**	**85.18**	98.93	**64.10**	71.82
w/o K/V grad.	26.15	84.64	99.07	51.30	**72.58**
w/o sink recache	24.42	81.70	**99.48**	6.90	71.24
SAF^{S}	25.99	**84.22**	99.06	**48.00**	72.23
w/o K/V grad.	**26.01**	84.02	99.07	44.50	**72.72**
w/o sink recache	24.21	81.30	**99.41**	4.30	71.04
Fixed750	**25.71**	**84.68**	98.94	**55.60**	71.93
Fixed500	25.40	84.12	99.02	47.10	72.13
Fixed250	25.65	83.99	**99.10**	45.70	**72.36**

Figure 4: **Quality across SDF training schedules on the first 20 seconds of MovieGen-100s.** The x-axis gives the random/fixed/aligned history ratio in training ($0/0/100$ is SAF; Appendix A). Bars are inference policies: Aligned uses stage-aligned multiple K/V banks (SAF^{M}), Mix one bank written at noise levels sampled from $(250, 500, 750)$ with probabilities $(0.5, 0.25, 0.25)$ (SAF^{S}), and Fixed250 one bank written at noise level 250.



while preserving visual fidelity; in contrast, LongLive can introduce semantic errors such as duplicating the cat, and SGF fails to realize the requested prompt-conditioned motion. On MovieGen-100s, ours maintains the table-wiping motion and scene layout over the full horizon, whereas SGF duplicates the human subject and LongLive produces inconsistent background reflections. Additional comparisons, including framewise results, are provided in Appendix F, and more visualizations are provided in the supplementary material.



Figure 5: **Chunkwise Interactive qualitative comparison.** Ours tracks the changing ten-second prompts with faithful semantics and stable visual quality, while competing methods exhibit quality degradation, object-count errors, or missed interactions.

4.3 Ablation Studies

We conduct ablations on 20-second subsets of MovieGen-100s and Interactive, with protocol details and full metrics reported in Appendix E. Table 2 shows consistent trends for both SAF^{M} and SAF^{S}. Detaching the history K/V mildly reduces Quality but substantially lowers Dynamic Degree, confirming the importance of the write term, i.e., future-to-history supervision. Removing sink recaching causes the largest degradation in both metrics. For single-bank inference under Fixed$t$, a larger t generally improves Quality and Dynamic Degree, but reduces Motion Smoothness and Imaging Quality, revealing a trade-off between dynamics and visual stability. We therefore adopt the Mix



Figure 6: **Chunkwise MovieGen-100s qualitative comparison.** Ours sustains the long-horizon table-wiping action with stable scene structure, whereas competing methods accumulate artifacts, duplicate subjects, or produce inconsistent reflections.

policy for SAF^{S}, which provides a more balanced operating point across dynamics, smoothness, and image fidelity.

To examine how the noise level of the history K/V affects training, Figure 4 compares schedules of Self-Diffusion Forcing (SDF), a two-pass generalization of SAF that writes each block's K/V at a random, fixed (t=250), or aligned stage (Appendix A). A schedule $a/b/c$ samples these three policies with probabilities $a\%$, $b\%$, and $c\%$, so $0/0/100$ is SAF. Across the evaluated schedules, aligned inference consistently achieves the highest Quality, while Mix is typically second-best. For the fixed-trained model, Fixed250 performs better, consistent with its matched training history distribution. SAF ($0/0/100$) achieves the highest Quality under every inference policy.

Training efficiency. SAF is the most efficient in both settings (Table 3). Block-parallel rollout reduces the sequential depth to the number of stages, and the supervised forward pass itself yields the differentiable history, removing the history-construction rollout and per-block recaching. The gain is largest in the framewise setting, where serial rollout is most costly: SAF trains $1.8\times$ faster than SGF and nearly matches its own chunkwise speed (137 s vs. 133 s). Even the two-pass SDF (Appendix A) is slightly faster than SGF, as it skips per-block timestep-zero recaching.

Table 3: **Training efficiency on one GB300** at batch size 4: memory (GiB) and time (s) per five-step cycle. Local attention spans 12 frames (chunkwise) and 21 frames (framewise). Best is **bold** and second-best underlined.

Method	Chunkwise		Framewise	
	Mem. ↓	Time ↓	Mem. ↓	Time ↓
LongLive	173.63	<u>158.32</u>	174.21	<u>232.64</u>
SGF	187.45	186.73	187.45	249.21
SDF	<u>160.85</u>	184.33	<u>171.30</u>	245.87
SAF	**155.18**	**132.78**	**164.79**	**137.04**

5 Conclusion

In this paper, we show that history K/V along the denoising trajectory, when aligned with the current noise level, can serve as an effective and trainable memory for streaming video diffusion. SAF realizes this by aligning each block's history with its denoising stage and rolling out all blocks at each stage in a single causal forward pass. This keeps the noisy history differentiable, allowing future losses to supervise how it is written, while avoiding both a separate no-gradient rollout to construct the history and per-block timestep-zero recaching. Empirically, SAF improves long-horizon generation and prompt switching, accelerates training by up to $1.8\times$ over SGF with lower memory, and supports efficient multi-GPU pipelined inference. Its noisy local history can still cause abrupt changes when new content enters the frame under camera motion (Appendix G). Overall, our results suggest that optimizing history directly along the denoising trajectory is a promising path toward efficient, long-horizon, and interactive video generation.

AI USE STATEMENT

Generative AI tools were used to assist with implementing the proposed method, as well as translation and language polishing of the manuscript. The research ideas originated from the authors. AI tools were not used to generate research data or analyze experimental results. The authors take responsibility for the final content of the paper and the implementation.

ETHICS STATEMENT

This work involves no human subjects or personal data; the generator is distilled from the public Wan2.1 models using only text prompts and evaluated on public prompt sets. Real-time, prompt-controllable video generation could be misused to fabricate events or to impersonate real people in live interactions, such as fake video calls or livestreams, which are harder to recognize as synthetic because they respond to viewers in real time. We advocate clear disclosure of synthetic content and the use of watermarking. Any release of our model will include usage restrictions prohibiting deceptive impersonation.

REPRODUCIBILITY STATEMENT

Section 3 and Appendices A and B specify SAF and SDF, including their training (Algorithms 1 and 5) and inference (Algorithms 2, 3, and 4) procedures. Appendix C.1 lists the models, hyperparameters, selected checkpoints, and hardware; Appendix C.2 describes the benchmarks which are publicly available; Appendix E details the ablation protocols. We will release our code, configurations, and model checkpoints upon publication.

REFERENCES

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C.Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando De Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative Interactive Environments. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/bruce24a.html.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token Prediction Meets Full-Sequence Diffusion. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 24081–24125. Curran Associates, Inc., 2024. doi: 10.52202/079017-0759. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/2aee1c4159e48407d68fe16ae8e6e49e-Paper-Conference.pdf.

Yuwei Guo, Ceyuan Yang, Hao He, Yang Zhao, Meng Wei, Zhenheng Yang, Weilin Huang, and Dahua Lin. End-to-End Training for Autoregressive Video Diffusion via Self-Resampling. In *Computer Vision – ECCV 2026*, Lecture Notes in Computer Science, pp. 324–344, Cham, 2026. Springer Nature Switzerland. doi: 10.1007/978-3-032-37574-2_18.

Xianglong He, Chunli Peng, Zexiang Liu, Boyang Wang, Yifan Zhang, Qi Cui, Fei Kang, Biao Jiang, Mengyin An, Yangyang Ren, Baixin Xu, Hao-Xiang Guo, Kaixiong Gong, Size Wu, Wei Li, Xuchen Song, Yang Liu, Yangguang Li, and Yahui Zhou. Matrix-Game 2.0: An Open-Source Real-Time and Streaming Interactive World Model, 2025. URL https://arxiv.org/abs/2508.13009.

Jonathan Ho and Tim Salimans. Classifier-Free Diffusion Guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the Train-Test Gap in Autoregressive Video Diffusion. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 167283–167308. Curran Associates, Inc., 2025. doi: 10.52202/085713-5576. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/f4823f831af67a3ef15e41a85434422a-Paper-Conference.pdf.

Yubo Huang, Hailong Guo, Fangtai Wu, Weiqiang Wang, Shifeng Zhang, Shijie Huang, Qijun Gan, Lin Liu, Sirui Zhao, Enhong Chen, Jiaming Liu, and Steven Hoi. Live Avatar: Streaming Real-time Audio-Driven Avatar Generation with Infinite Length, 2026a. URL https://arxiv.org/abs/2512.04677.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive Benchmark Suite for Video Generative Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, June 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*, 48(3):3268–3285, 2026b. doi: 10.1109/TPAMI.2025.3633890.

Sihui Ji, Xi Chen, Shuai Yang, Xin Tao, Pengfei Wan, and Hengshuang Zhao. MemFlow: Flowing Adaptive Memory for Consistent and Efficient Long Video Narratives, 2025. URL https://arxiv.org/abs/2512.14699.

Akio Kodaira, Tingbo Hou, Ji Hou, Markos Georgopoulos, Felix Juefei-Xu, Masayoshi Tomizuka, and Yue Zhao. StreamDiT: Real-Time Streaming Text-to-Video Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29200–29210, June 2026.

Shanchuan Lin, Ceyuan Yang, Hao He, Jianwen Jiang, Yuxi Ren, Xin Xia, Yang Zhao, Xuefeng Xiao, and Lu Jiang. Autoregressive Adversarial Post-Training for Real-Time Interactive Video Generation. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 41061–41086. Curran Associates, Inc., 2025. doi: 10.52202/085713-1371. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/3a9468a918fc65dc9ce7b7bd99f4f0ef-Paper-Conference.pdf.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling Forcing: Autoregressive Long Video Diffusion in Real Time. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IAyzXjbfwo.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *The Seventh International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Yunhong Lu, Yanhong Zeng, Haobo Li, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jiapeng Zhu, Hengyuan Cao, Zhipeng Zhang, Xing Zhu, Yujun Shen, and Min Zhang. Reward Forcing: Efficient Streaming Video Generation with Rewarded Distribution Matching Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 34385–34397, June 2026.

Ryan Po, Eric Ryan Chan, Changan Chen, and Gordon Wetzstein. BAgger: Backwards Aggregation for Mitigating Drift in Autoregressive Video Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 43727–43739, June 2026.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, David Yan, Dhruv Choudhary, Dingkang Wang, Geet Sethi, Guan Pang, Haoyu Ma, Ishan Misra, Ji Hou, Jialiang Wang, Kiran Jagadeesh, Kunpeng Li, Luxin Zhang, Mannat Singh, Mary Williamson, Matt Le, Matthew Yu, Mitesh Kumar Singh, Peizhao Zhang, Peter Vajda, Quentin Duval, Rohit Girdhar, Roshan Sumbaly, Sai Saketh Rambhatla, Sam Tsai, Samaneh Azadi, Samyak Datta, Sanyuan Chen, Sean Bell, Sharadh Ramaswamy, Shelly Sheynin, Siddharth Bhattacharya, Simran Motwani, Tao Xu, Tianhe Li, Tingbo Hou, Wei-Ning Hsu, Xi Yin, Xiaoliang Dai, Yaniv Taigman, Yaqiao Luo, Yen-Cheng Liu, Yi-Chiao Wu, Yue Zhao, Yuval Kirstain, Zecheng He, Zijian He, Albert Pumarola, Ali Thabet, Artsiom Sanakoyeu, Arun Mallya, Baishan Guo, Boris Araya, Breena Kerr, Carleigh Wood, Ce Liu, Cen Peng, Dimitry Vengertsev, Edgar Schonfeld, Elliot Blanchard, Felix Juefei-Xu, Fraylie Nord, Jeff Liang, John Hoffman, Jonas Kohler, Kaolin Fire, Karthik Sivakumar, Lawrence Chen, Licheng Yu, Luya Gao, Markos Georgopoulos, Rashel Moritz, Sara K. Sampson, Shikai Li, Simone Parmeggiani, Steve Fine, Tara Fowler, Vladan Petrovic, and Yuming Du. Movie Gen: A Cast of Media Foundation Models, 2025. URL https://arxiv.org/abs/2410.13720.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence Level Training with Recurrent Neural Networks. In *The Fourth International Conference on Learning Representations*, 2016. URL https://arxiv.org/abs/1511.06732.

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

Dani Valevski, Yaniv Leviathan, Moab Arar, and Shlomi Fruchter. Diffusion Models Are Real-Time Game Engines. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=P8pqeEkn1H.

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, 2025. URL https://arxiv.org/abs/2503.20314.

Wenhao Wang and Yi Yang. VidProM: A Million-scale Real Prompt-Gallery Dataset for Text-to-Video Diffusion Models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 65618–65642. Curran Associates, Inc., 2024. doi: 10.52202/079017-2096. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/78b3e7836e3b7dea79d809b0c99cb097-Paper-Datasets_and_Benchmarks_Track.pdf.

Yi Wang, Yinan He, Yizhuo Li, Kunchang Li, Jiashuo Yu, Xin Ma, Xinhao Li, Guo Chen, Xinyuan Chen, Yaohui Wang, Ping Luo, Ziwei Liu, Yali Wang, Limin Wang, and Yu Qiao. InternVid: A Large-scale Video-Text Dataset for Multimodal Understanding and Generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=MLBdiWu4Fw.

Zhuguanyu Wu, Ruihao Gong, Yang Yong, Yushi Huang, Xiangyu Fan, Lei Yang, Dahua Lin, and Xianglong Liu. SGMD: Score Gradient Matching Distillation for Few-Step Video Diffusion Distillation. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=melQSx2bVT.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient Streaming Language Models with Attention Sinks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NG7sS51zVF.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Ying-Cong Chen, Yao Lu, Song Han, and Yukang Chen. LongLive: Real-time Interactive Long Video Generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=nCAODkpsPJ.

Jung Yi, Wooseok Jang, Paul Hyunbin Cho, Jisu Nam, Heeji Yoon, and Seungryong Kim. Deep Forcing: Training-Free Long Video Generation with Deep Sink and Participative Compression. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=gtmyFnvXAW.

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Frédo Durand, and William T. Freeman. Improved Distribution Matching Distillation for Fast Image Synthesis. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 47455–47487. Curran Associates, Inc., 2024a. doi: 10.52202/079017-1505. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/54dcf25318f9de5a7a01f0a4125c541e-Paper-Conference.pdf.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step Diffusion with Distribution Matching Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6613–6623, June 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 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22963–22974, June 2025.

Yuyang You, Yongzhi Li, Jiahui Li, Yadong Mu, Quan Chen, and Peng Jiang. Adaptive Video Distillation: Mitigating Oversaturation and Temporal Collapse in Few-Step Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 43429–43439, June 2026.

Yonghao Yu, Lang Huang, Runyi Li, Zerun Wang, and Toshihiko Yamasaki. Video-Mirai: Autoregressive Video Diffusion Models Need Foresight, 2026. URL https://arxiv.org/abs/2606.03971.

Lin Zhang, Sicheng Mo, Zefan Cai, Jinhong Lin, Zihao Lin, Jiuxiang Gu, Krishna Kumar Singh, Yuheng Li, and Yin Li. UniTemp: Unlocking Video Generation in Any Temporal Order via Bidirectional Distillation, 2026. URL https://arxiv.org/abs/2606.18702.

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. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=BYInOck3gr.

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

Kai Zou, Dian Zheng, Hongbo Liu, Tiankai Hang, Bin Liu, and Nenghai Yu. HiAR: Efficient Autoregressive Long Video Generation via Hierarchical Denoising. In *Computer Vision – ECCV 2026*, Lecture Notes in Computer Science, pp. 225–242, Cham, 2026. Springer Nature Switzerland. doi: 10.1007/978-3-032-37026-6_13.

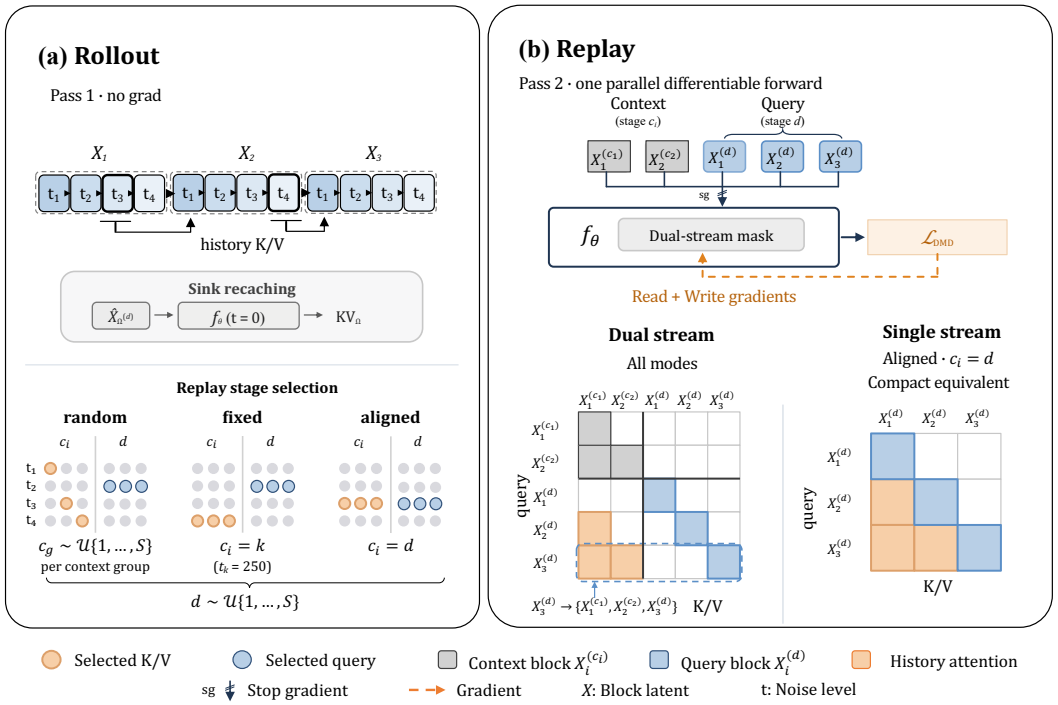


Figure 7: **Two-pass training of SDF. (a)** Pass 1 rolls out all blocks serially without gradients, recaches the sink from $\widehat{X}_{\Omega}^{(d)}$, and records the trajectory; the history of each block is read at the stage chosen by the random, fixed, or aligned policy. **(b)** Pass 2 replays the recorded states in one differentiable forward pass over a context stream C and a query stream D, so the DMD loss on the queries reaches the context K/V they attend to. Under the aligned policy ($c_i = d$), context and query coincide and the two streams collapse into one. For clarity, the attention masks omit the clean sink; as in SAF, block i also attends to the clean K/V of the preceding sink blocks, $\mathrm{KV}_{\Omega,<i}$ (Section 3.4). At the top of (a), X_i is block i, whose K/V enters the history at the bold stage; the context and query streams in (b) hold its recorded states $X_i^{(c_i)}$ and $X_i^{(d)}$, and c_g is shared by context group g (Eq. 7).

A Self-Diffusion Forcing

SAF encodes the history of every block at the stage of the current block. To test this choice, our schedule ablations (Section 4.3 and Appendix E.2) use Self-Diffusion Forcing (SDF), a generalization that decouples the stage c_i at which block i writes its K/V into the history from the query stage $d \sim \mathcal{U}\{1, \dots, S\}$ at which predictions are supervised. In SAF, d is also the exit stage at which the rollout stops (Section 3.3); in SDF, it does not truncate the rollout: Pass 1 runs all S stages, and d only selects which recorded states are replayed with gradients. Figure 7 illustrates the procedure.

History-stage policies. SDF supports three policies for the history stage:

$$c_i = d \text{ (aligned)}, \qquad c_i = k \text{ (fixed)}, \qquad c_i = c_{g(i)},\ c_g \sim \mathcal{U}\{1, \dots, S\} \text{ (random)}, \tag{7}$$

where k is the stage with $t_k = 250$ and $g(i)$ is the group of three latent frames containing block i. A training schedule draws one policy per sample, and schedule $a/b/c$ uses the random, fixed, and aligned policies with probabilities $a\%$, $b\%$, and $c\%$; the $0/0/100$ schedule is SAF.

Two-pass training. When $c_i \neq d$, the history and the query of a block lie at different noise levels and cannot share a forward pass, so SDF trains in two passes. As in SAF, the sink blocks are first rolled out sequentially, and their estimate $\widehat{X}_{\Omega}^{(d)}$ at the query stage d serves as the clean sink recached in both passes. Pass 1 rolls out the remaining blocks serially without gradients, reading the history at the policy stage, and records their states. Pass 2 packs the clean sink, a context stream C with $C_i = X_i^{(c_i)}$, and a query stream D with $D_i = X_i^{(d)}$, all recorded in Pass 1, into one differentiable forward pass. Under the dual-stream mask M_{dual}, query block i attends to the clean sink blocks

Algorithm 3 Single-GPU Multi-Bank Inference of SAF^{M}

Require: Denoiser f_θ, stages $t_{1:S}$, sampler Ψ, N blocks, window L, sink Ω, block prompts $p_{1:N}$
Ensure: Ordered clean blocks $\widehat{X}_{1:N}^{(S)}$

Disable gradients; generate and emit the sink blocks sequentially
Encode $\widehat{X}_{\Omega}^{(S)}$ once at $t = 0$ to obtain KV_Ω
$\mathcal{H}^{(s)} \leftarrow \mathrm{Init}(\mathrm{KV}_\Omega)$ for $s = 1, \ldots, S$
for $i = |\Omega| + 1, \ldots, N$ **do**
    Sample $X_i^{(1)} \sim \mathcal{N}(0, I)$
    for $s = 1, \ldots, S$ **do**
        $(\widehat{X}_i^{(s)}, \mathrm{KV}_i^{(s)}) \leftarrow f_\theta(X_i^{(s)}, t_s; \mathcal{H}^{(s)}, p_i)$
        $\mathcal{H}^{(s)} \leftarrow \mathrm{Commit}_L(\mathcal{H}^{(s)}, \mathrm{KV}_i^{(s)})$
        if $s < S$ **then**
            Sample $\epsilon_{i,s} \sim \mathcal{N}(0, I)$; $X_i^{(s+1)} \leftarrow \Psi_s(\widehat{X}_i^{(s)}, \epsilon_{i,s})$
        else
            Emit $\widehat{X}_i^{(S)}$
        end if
    end for
end for

before it, the context blocks $j < i$ within the window, and its own query tokens, but never to its own context copy, other query blocks, or future context. Only query predictions receive the DMD loss, and its gradients reach the sink and context K/V that the queries attend to, but not the recorded trajectory. Under the aligned policy, C and D coincide and M_{dual} reduces to M_{causal}. SAF goes one step further: because every stage produces its own history, it replaces the serial Pass 1 with the block-parallel rollout of Section 3.3. Algorithm 5 in Appendix B gives the full procedure.

B Algorithm Details

This section complements Algorithms 1 and 2 with the shared cache operations, single-GPU inference for SAF^{M} and SAF^{S} (Algorithms 3 and 4), and the two-pass SDF training of Appendix A (Algorithm 5). Notation follows Section 3.

Denoiser and sampling. $\mathrm{KV}_i^{(s)}$ collects the K/V of block i from all attention layers. In the training algorithms, f_θ processes all blocks in one forward pass under the indicated attention mask, whereas at inference it processes one block against a history K/V bank. Gaussian draws are indexed by sample, block, and stage, so all execution schedules of the same sample use identical noise. Blocks keep their original temporal indices for positional encoding. Training uses a single prompt p, whereas at inference block i uses its own prompt p_i, so a prompt switch changes only the text conditioning and leaves the generated history intact.

Cache operations. The sink Ω consists of the leading blocks of the video, and *sink recaching* encodes their clean latents at $t = 0$ into the clean K/V KV_Ω. During training, the sink is rolled out to the exit stage d, and the parallel forward pass of every stage recaches it from the same estimate $\widehat{X}_{\Omega}^{(d)}$ (Eq. 5); SDF likewise uses $\widehat{X}_{\Omega}^{(d)}$ (Appendix A). At inference, the sink blocks are generated first and recached once from their final estimate $\widehat{X}_{\Omega}^{(S)}$, after which the remaining blocks are streamed. $\mathrm{Init}(\mathrm{KV}_\Omega)$ creates a bank that holds KV_Ω and an empty local window, and all stage banks share the same sink. $\mathrm{Commit}_L(\mathcal{H}, \mathrm{KV})$ appends the K/V of the current block to $\mathcal{H}$ after its forward pass and evicts the oldest non-sink entries beyond the window L; the sink is never evicted. Apart from sink recaching, no procedure below runs a forward pass at $t = 0$.

Single-GPU multi-bank inference. Algorithm 3 runs SAF^{M} on one GPU by denoising each block through all S stages before starting the next. It respects the same dependencies as the pipeline of Algorithm 2, so both produce identical videos from the same noise: when block i reaches stage s, bank $\mathcal{H}^{(s)}$ holds exactly its history $\mathcal{H}_i^{(s)}$.

Algorithm 4 Single-Bank Inference of SAF^{S}

Require: Denoiser f_θ, stages $t_{1:S}$, sampler Ψ, N blocks, window L, sink Ω, block prompts $p_{1:N}$
Require: Commit policy π over stages $\{1, \dots, S\}$ (Fixed$t$ or Mix)
Ensure: Ordered clean blocks $\widehat{X}_{1:N}^{(S)}$

Disable gradients; generate and emit the sink blocks sequentially
Encode $\widehat{X}_{\Omega}^{(S)}$ once at $t = 0$ to obtain KV_Ω
$\mathcal{H} \leftarrow \mathrm{Init}(\mathrm{KV}_\Omega)$
for $i = |\Omega| + 1, \dots, N$ **do**
    Sample $c_i \sim \pi$ and $X_i^{(1)} \sim \mathcal{N}(0, I)$; $\mathrm{KV}_i^\star \leftarrow \emptyset$
    for $s = 1, \dots, S$ **do**
        $(\widehat{X}_i^{(s)}, \mathrm{KV}_i^{(s)}) \leftarrow f_\theta(X_i^{(s)}, t_s; \mathcal{H}, p_i)$ ▷ $\mathcal{H}$ is read-only
        if $s = c_i$ **then**
            $\mathrm{KV}_i^\star \leftarrow \mathrm{KV}_i^{(s)}$ ▷ keep the K/V of stage c_i
        end if
        if $s < S$ **then**
            Sample $\epsilon_{i,s} \sim \mathcal{N}(0, I)$; $X_i^{(s+1)} \leftarrow \Psi_s(\widehat{X}_i^{(s)}, \epsilon_{i,s})$
        end if
    end for
    $\mathcal{H} \leftarrow \mathrm{Commit}_L(\mathcal{H}, \mathrm{KV}_i^\star)$; emit $\widehat{X}_i^{(S)}$
end for

Single-bank inference. Algorithm 4 keeps one bank $\mathcal{H}$ shared by all stages. For each block, it samples the stage $c_i \sim \pi$ whose K/V is committed, where π places all mass on the stage with noise level t for Fixed$t$, and for Mix samples the stages with noise levels $(250, 500, 750)$ with probabilities $(0.5, 0.25, 0.25)$. All stages of block i read the same history, and only the K/V from stage c_i is committed once the block is fully denoised. The bank thus holds one entry per block, and different blocks may be written at different noise levels.

Two-pass SDF training. Algorithm 5 implements the two passes of Appendix A; score-model updates follow the standard DMD procedure. A context-stage plan $\{c_j\} \sim \pi(\cdot \mid d)$, drawn according to Eq. 7, is shared by both passes. In Pass 1, $\mathrm{SelectHistory}$ reads the preceding blocks at stage k under the fixed policy, at the planned stages under the random policy, and at the current stage s under the aligned policy, while $\mathrm{CommitPlan}_L$ retains only the K/V that the policy requires. Both passes share the clean sink recached from $\widehat{X}_{\Omega}^{(d)}$, and the recorded trajectory $\mathcal{R}$ receives no gradients.

C Experimental Details

C.1 Implementation

Distillation setup. The generator is a causal Wan2.1-T2V-1.3B (Wan et al., 2025) trained with DMD (Yin et al., 2024b;a). Wan2.1-T2V-14B serves as the real score model with a classifier-free guidance (Ho & Salimans, 2022) scale of 3.0, and the bidirectional Wan2.1-T2V-1.3B serves as the fake score model. Following SGF (Zhuang et al., 2026), we initialize SAF and SDF from the `ar_diffusion` checkpoint released by Causal-Forcing (Zhu et al., 2026). Training uses text prompts only, taken from the VidProM prompts (Wang & Yang, 2024) as filtered and extended by Self-Forcing (Huang et al., 2025). Each training sample has 21 latent frames, corresponding to a 5-second video at 832×480, 16 FPS. All models use $S = 4$ denoising stages with noise levels $(t_1, \dots, t_4) = (1000, 750, 500, 250)$ and a timestep shift of 5.

Attention. In the *chunkwise* setting, each block contains 3 latent frames, and the local attention window spans 12 latent frames, including the current block and a 3-frame sink. In the *framewise* setting, each block contains 1 latent frame, and the window spans 21 latent frames with a 4-frame sink, following SGF (Zhuang et al., 2026). The window works as a first-in, first-out queue, except for the sink, which stays fixed. The sink holds the latent frames at the start of the video and is recached at timestep zero from a single estimate, taken at the exit stage during training and at the final stage at inference, whereas the rest of the history keeps the noisy K/V from the denoising trajectory (Appendix B).

Algorithm 5 Two-Pass Training of SDF

Require: Denoiser f_θ, stages $t_{1:S}$, sampler Ψ, N blocks, window L, sink Ω, prompt p
Require: History policy π, dual-stream causal mask M_{dual}
Ensure: One generator update

Sample $d \sim \mathcal{U}\{1, \dots, S\}$ and context-stage plan $\{c_j\} \sim \pi(\cdot \mid d)$
Pass 1: detached trajectory collection
Disable gradients; roll out the sink blocks sequentially; record $X_\Omega^{(1:S)}$ and $\widehat{X}_\Omega^{(d)}$
Encode $\widehat{X}_\Omega^{(d)}$ at $t = 0$ into KV_Ω ▷ sink recaching
$\mathcal{H} \leftarrow \mathrm{Init}(\mathrm{KV}_\Omega)$; $\mathcal{R}_{i,s} \leftarrow X_i^{(s)}$ for $i \in \Omega$ and all s
for $i = |\Omega| + 1, \dots, N$ **do**
  Sample $X_i^{(1)} \sim \mathcal{N}(0, I)$
  for $s = 1, \dots, S$ **do**
    $\mathcal{R}_{i,s} \leftarrow \mathrm{sg}(X_i^{(s)})$
    $\mathcal{H}_i^{(s)} \leftarrow \mathrm{SelectHistory}(\mathcal{H}, \pi, \{c_j\}_{j<i}, s)$
    $(\widehat{X}_i^{(s)}, \mathrm{KV}_i^{(s)}) \leftarrow f_\theta(X_i^{(s)}, t_s; \mathcal{H}_i^{(s)}, p)$
    if $s < S$ **then**
      Sample $\epsilon_{i,s} \sim \mathcal{N}(0, I)$; $X_i^{(s+1)} \leftarrow \Psi_s(\widehat{X}_i^{(s)}, \epsilon_{i,s})$
    end if
  end for
  $\mathcal{H} \leftarrow \mathrm{CommitPlan}_L(\mathcal{H}, \mathrm{KV}_i^{(1:S)}, \pi, c_i)$
end for
Discard the rollout cache $\mathcal{H}$
Pass 2: differentiable replay
$C_i \leftarrow \mathcal{R}_{i,c_i}$ and $D_i \leftarrow \mathcal{R}_{i,d}$ for $i = 1, \dots, N$
Pack $Z \leftarrow [\widehat{X}_\Omega^{(d)} \mid C \mid D]$ and timestep vector $\mathbf{t} \leftarrow [0 \mid \{t_{c_i}\} \mid t_d]$; enable gradients
$\widehat{X}_D \leftarrow [f_\theta(\mathrm{sg}(Z), \mathbf{t}; M_{\mathrm{dual}}, p)]_D$ ▷ clean sink recached with gradients
$\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{DMD}}(\widehat{X}_D, p)$
Backpropagate through sink, context, and query K/V, but not into $\mathcal{R}$ or Pass 1; update θ

Optimization. We use AdamW (Loshchilov & Hutter, 2019) with $(\beta_1, \beta_2) = (0, 0.999)$ and a weight decay of 0.01. The learning rate is 2×10^{-6} for the generator and 4×10^{-7} for the fake score model. The fake score model is first warmed up for 10 updates, after which the generator is updated once every 5 fake score updates. We train for 800 steps on 8 NVIDIA GB300 GPUs with a batch size of 4 per GPU (32 in total) and a random seed of 42. Starting at step 200, we also track an exponential moving average (EMA) of the generator weights with a decay of 0.99.

Inference. We evaluate the EMA weights at step 600 for the chunkwise model and at step 700 for the framewise model. $\mathrm{SAF}^{\mathrm{M}}$ maintains a separate K/V bank for each denoising stage, whereas $\mathrm{SAF}^{\mathrm{S}}$ maintains a single bank written by the Mix policy (Appendix B). Inference uses the base seed 42, from which per-sample seeds are derived as described in Appendix C.2.

C.2 Benchmarks and Metrics

Table 4 summarizes the three benchmarks. Seeding is deterministic in all evaluations. The seed of each sample is derived from the base seed 42, its global prompt index, and its sample index, so the results do not depend on how prompts are split across GPUs or on which subset is evaluated.

Table 4: **Evaluation benchmarks.** All models are trained on 5-second clips, so Interactive and MovieGen-100s test generation at $12\times$ and $20\times$ the training length. Each Interactive video is driven by six prompts in turn, each lasting 10 seconds, and the benchmark provides 100 such prompt sequences (100×6). Samples is the number of videos per prompt (per prompt sequence for Interactive), and Dimensions is the number of VBench dimensions scored.

Benchmark	Prompts	Samples	Length	Dimensions	Aggregate metrics
VBench	944	5	5 s	16	Total, Quality, Semantic
Interactive	100×6	1	60 s	7	Quality, ViCLIP
MovieGen-100s	128	1	100 s	7	Quality

Table 5: **Framewise evaluation with existing methods.** Framewise counterpart of Table 1; Causal-Forcing and SGF are the only baselines with released framewise checkpoints. Throughput is reported as in Table 1, with the second value denoting 4-GPU pipelined inference. Best results are **bold** and second-best results are underlined.

Method	Throughput (FPS) ↑	VBench			Interactive		MovieGen 100s
		Total ↑	Quality ↑	Semantic ↑	ViCLIP ↑	Quality ↑	Quality ↑
Causal-Forcing	8.9	83.86	85.23	<u>78.35</u>	21.82	81.42	80.62
SGF	8.9	**84.10**	<u>85.26</u>	**79.48**	<u>24.85</u>	**84.83**	**84.71**
SAF^{S}	<u>9.07</u>	83.20	84.52	77.93	22.85	81.95	80.77
SAF^{M}	**9.08 / 21.6**	<u>83.88</u>	**85.31**	78.15	**25.23**	<u>84.35</u>	<u>84.17</u>

Protocols. All three benchmarks follow existing evaluation protocols. On **VBench** (Huang et al., 2024; 2026b), we generate five videos for each of the 944 unique standard prompts and report the Total, Quality, and Semantic scores. **Interactive** uses the 100 prompt sequences released by MemFlow (Ji et al., 2025) and the protocol of LongLive (Yang et al., 2026). Each 60-second video is generated from six prompts in turn, one for every 10 seconds. Prompt alignment is the ViCLIP (Wang et al., 2024) similarity between each 10-second segment and its own prompt, averaged over the six segments. **MovieGen-100s** uses the 128 extended MovieGen (Polyak et al., 2025) prompts released by UniTemp (Zhang et al., 2026) and generates one 100-second video per prompt. On Interactive and MovieGen-100s, Quality is computed with VBench-Long (Huang et al., 2026b), which splits each video into 2-second clips and scores them on the same seven quality dimensions as VBench. All per-dimension scores in the tables are multiplied by 100.

C.3 Baselines and Throughput

We re-evaluate Self-Forcing (Huang et al., 2025), LongLive (Yang et al., 2026), Causal-Forcing (Zhu et al., 2026), HiAR (Zou et al., 2026), and SGF (Zhuang et al., 2026) with their official weights and released inference settings, using the prompts and seeds described above. Only Causal-Forcing and SGF release framewise checkpoints, so the framewise comparison (Table 5) includes only these two baselines. Baseline throughput is taken from previously reported results. For SAF, we measure throughput on H100 GPUs both on a single GPU and with 4-GPU pipelined inference (Algorithm 2), reported as the first and second values, respectively.

D Additional Quantitative Results

Framewise comparison. Table 5 is the framewise counterpart of Table 1. SAF^{M} has the highest throughput, VBench Quality, and Interactive ViCLIP. SGF leads in VBench Total and Semantic, as well as in Quality on Interactive and MovieGen-100s. SAF^{M} outperforms SAF^{S} on all six aggregate scores, and the gap between the two inference policies is larger than in the chunkwise setting (e.g., 3.40 versus 0.66 in MovieGen-100s Quality). We attribute this gap to error accumulation in single-bank inference, which is stronger in the framewise setting: each block contains a single latent frame that cannot be refined jointly with its neighbors, and the larger 21-frame window keeps more of this history in context (Appendix G).

Per-dimension results. Tables 6–8 break down the aggregate scores in Tables 1 and 5. Consistency and motion generally trade off against each other. Causal-Forcing has the highest dynamic degree in five of the six benchmark–setting pairs but the lowest background consistency in all six. SGF shows the opposite pattern in the framewise setting, with the highest background consistency and the lowest dynamic degree on all three benchmarks.

- **VBench** (Table 6). In the chunkwise setting, SAF^{M} is best in temporal flickering (tied with SAF^{S}) and object class, and second in dynamic degree, subject consistency, and background consistency. Other methods are stronger in color, spatial relationship, and the style dimensions. In the framewise setting, SAF^{M} leads in motion smoothness, aesthetic quality, and object class, while SGF leads in subject consistency, background consistency, and imaging quality.

Table 6: **VBench per-dimension results.** All 16 dimensions of the 5-second evaluation; higher is better. Chunkwise (left) and framewise (right) results are ranked separately; best and second-best entries are **bold** and underlined.

Metric	Chunkwise							Framewise			
	Self-Forcing	LongLive	Causal-Forcing	HiAR	SGF	**SAF**^{S}	**SAF**^{M}	Causal-Forcing	SGF	**SAF**^{S}	**SAF**^{M}
Subject Consistency	95.09	96.98	95.48	96.35	**97.14**	96.25	97.05	91.01	**96.91**	95.48	96.28
Background Consistency	96.10	**96.92**	95.86	96.31	96.35	96.58	96.62	92.85	**96.02**	94.84	95.14
Temporal Flickering	99.01	99.35	99.22	99.18	99.07	**99.54**	**99.54**	**99.48**	99.11	98.66	98.71
Motion Smoothness	98.24	**98.79**	97.53	98.35	98.51	98.46	98.39	97.25	98.23	98.51	**98.54**
Dynamic Degree	66.38	40.83	**84.44**	47.22	66.94	71.39	76.39	**99.44**	64.72	66.94	72.78
Aesthetic Quality	65.79	67.03	66.63	66.51	**68.32**	67.68	67.83	65.44	67.21	66.83	**67.30**
Imaging Quality	69.71	69.18	70.31	67.92	**70.58**	69.04	68.88	69.02	**71.20**	69.22	69.38
Object Class	93.16	96.28	95.17	95.21	95.63	96.49	**96.61**	93.89	95.52	95.70	**95.87**
Multiple Objects	87.19	86.49	**87.82**	82.93	86.83	85.27	84.82	75.63	**84.91**	81.43	82.26
Human Action	96.40	95.80	96.00	**96.60**	96.40	95.20	96.00	**96.00**	95.60	95.40	**96.00**
Color	86.83	**90.79**	86.16	89.04	88.98	88.10	88.38	83.92	**87.75**	83.29	83.79
Spatial Relationship	**81.77**	80.56	80.98	77.33	80.32	77.67	78.22	**74.49**	74.33	70.62	71.37
Scene	56.13	**58.79**	55.00	53.85	56.13	55.19	55.17	55.36	**55.74**	54.68	54.42
Appearance Style	20.34	20.42	20.53	**20.56**	20.46	20.21	20.20	**20.72**	20.46	20.43	20.40
Temporal Style	24.45	24.16	**24.72**	24.42	24.20	24.03	23.97	**25.06**	23.90	23.73	23.57
Overall Consistency	**26.85**	26.61	26.65	**26.85**	26.56	26.29	26.32	**26.41**	26.26	26.12	26.12

- **Interactive** (Table 7). In the chunkwise setting, SAF^{M} reaches a dynamic degree of 73.97, well above the best baseline (56.20). From the second segment onward, its ViCLIP is second only to LongLive, whereas the ViCLIP of Self-Forcing declines after every prompt switch. In the framewise setting, SAF^{M} achieves the best ViCLIP in every segment after the first.
- **MovieGen-100s** (Table 8). In the chunkwise setting, SAF^{M} is best in subject consistency (tied with SAF^{S}), background consistency, and aesthetic quality, and nearly matches the dynamic degree of Causal-Forcing (65.61 versus 65.69). In the framewise setting, SAF^{M} leads in subject consistency and aesthetic quality. SAF^{S} keeps strong motion but has the lowest imaging quality, consistent with the drift discussed in Appendix G.

E Ablation Details

E.1 Component and K/V Bank Ablations

Setup. Tables 2 and 9 ablate two training components of SAF and the K/V bank used at inference, all in the chunkwise setting. *w/o K/V grad.* detaches the history K/V during training. Every block is still supervised, but gradients from later blocks no longer flow into the K/V of earlier blocks. *w/o sink recache* skips the timestep-zero recache of the sink, so the sink keeps stage-aligned noisy K/V like the rest of the history. Both training ablations are evaluated with the same SAF^{M} and SAF^{S} inference policies as the full model. Fixed$t$ instead runs the full SAF checkpoint with a single bank written only at noise level t. For each Interactive prompt sequence, we generate only the first 20 seconds, which cover the first two prompts, and score them with VBench-Long as in Appendix C.2. ViCLIP is averaged over the two 10-second segments.

Results. Table 9 extends Table 2 to all seven quality dimensions and both ViCLIP segments. Removing K/V gradients lowers dynamic degree under both inference policies, from 64.10 to 51.30 for SAF^{M} and from 48.00 to 44.50 for SAF^{S}. The slightly higher consistency and imaging quality come with this reduced motion. Removing sink recaching leads to nearly static videos. Dynamic degree falls below 7, while subject consistency, background consistency, and motion smoothness reach their highest values within each group. Second-segment ViCLIP also drops by more than 3 points, showing that the model fails to follow the new prompt. Under Fixed$t$, a larger t monotonically increases Quality and dynamic degree but lowers motion smoothness and imaging quality. The

Table 7: **Interactive per-dimension results.** The 7 VBench-Long quality dimensions (top) and ViCLIP prompt alignment for each 10-second segment (bottom); ViCLIP averages the six segments and equals the Interactive ViCLIP score in Tables 1 and 5. Chunkwise and framewise results are ranked separately; best and second-best entries are **bold** and underlined.

Metric	Chunkwise							Framewise			
	Self-Forcing	LongLive	Causal-Forcing	HiAR	SGF	**SAF**^{S}	**SAF**^{M}	Causal-Forcing	SGF	**SAF**^{S}	**SAF**^{M}
Subject Consistency	97.38	97.74	95.14	97.69	97.75	**97.87**	97.17	92.90	**97.70**	95.71	97.05
Background Consistency	96.42	96.58	95.08	96.12	96.55	**96.70**	96.17	94.14	**96.64**	95.16	95.60
Temporal Flickering	99.07	99.04	98.47	98.34	98.68	99.13	**99.15**	97.35	**99.02**	98.48	98.58
Motion Smoothness	99.10	**99.16**	97.68	98.66	98.97	99.00	98.83	97.77	**98.86**	97.61	98.62
Dynamic Degree	22.73	27.47	56.20	26.43	48.23	50.87	**73.97**	**86.43**	64.77	75.13	69.67
Aesthetic Quality	57.77	60.82	57.30	57.74	**61.65**	61.60	61.42	52.02	60.00	54.72	**60.32**
Imaging Quality	68.04	71.45	69.00	71.54	72.66	**72.74**	71.90	64.17	**71.92**	63.44	70.24
ViCLIP	21.88	**25.74**	24.21	24.16	24.63	24.70	25.34	21.82	24.85	22.85	**25.23**
ViCLIP Segment 1	27.12	27.48	**27.51**	27.25	27.14	27.10	27.06	**27.53**	26.70	26.63	26.53
ViCLIP Segment 2	24.54	**26.32**	25.57	23.46	24.62	24.88	26.01	21.71	25.10	22.97	**25.25**
ViCLIP Segment 3	22.98	**25.49**	23.76	23.72	24.18	24.44	25.24	19.44	24.69	22.56	**25.05**
ViCLIP Segment 4	20.11	**24.95**	23.38	23.17	23.55	23.67	24.27	21.41	23.88	22.04	**24.37**
ViCLIP Segment 5	19.15	**25.33**	22.30	23.73	24.28	24.10	24.78	18.95	24.47	21.68	**24.82**
ViCLIP Segment 6	17.38	**24.89**	22.74	23.64	24.00	24.00	24.68	21.88	24.25	21.19	**25.35**

Table 8: **MovieGen-100s per-dimension results.** The 7 VBench-Long quality dimensions over the full 100-second videos. Chunkwise and framewise results are ranked separately; best and second-best entries are **bold** and underlined.

Metric	Chunkwise							Framewise			
	Self-Forcing	LongLive	Causal-Forcing	HiAR	SGF	**SAF**^{S}	**SAF**^{M}	Causal-Forcing	SGF	**SAF**^{S}	**SAF**^{M}
Subject Consistency	96.64	97.64	95.06	97.53	98.02	**98.06**	**98.06**	92.98	97.70	95.05	**97.88**
Background Consistency	96.34	96.65	95.49	96.23	96.81	96.86	**96.93**	94.60	**96.63**	94.98	96.49
Temporal Flickering	98.52	**98.99**	96.02	98.36	98.48	98.67	98.73	96.28	**98.70**	97.70	98.24
Motion Smoothness	98.24	**98.68**	95.83	98.36	98.60	98.45	98.26	96.79	**98.25**	96.10	98.00
Dynamic Degree	28.91	44.44	**65.69**	39.80	58.15	53.95	65.61	**85.16**	67.10	80.79	67.82
Aesthetic Quality	53.20	61.41	56.97	58.74	64.00	64.03	**64.59**	52.50	62.42	55.57	**63.02**
Imaging Quality	62.23	68.48	66.08	70.51	71.66	**71.80**	70.09	64.70	**70.55**	60.35	68.14

Mix policy of SAF^{S} falls between Fixed750 and Fixed500 in Quality and dynamic degree, and between Fixed500 and Fixed250 in motion smoothness and imaging quality. It also achieves a higher ViCLIP score than any fixed bank (25.99 versus at most 25.71).

E.2 Training Schedule Ablations

Setup. Figure 4 and Table 10 compare SDF training schedules $a/b/c$, which sample the random, fixed, and aligned policies of Eq. 7 with probabilities $a\%$, $b\%$, and $c\%$ (Appendix A). SAF is trained in a single pass (Algorithm 1), and the other schedules use the two-pass SDF procedure (Algorithm 5). All schedules share the same initialization and hyperparameters. Each model generates a 20-second video for each of the 128 MovieGen-100s prompts, scored with VBench-Long as above.

Results. Table 10 reports all dimensions behind Figure 4. Under every training schedule, Aligned inference gives the highest Quality and dynamic degree. SAF also achieves the highest Quality under every inference policy (83.97, 84.59, and 85.22 for Fixed250, Mix, and Aligned), and with Aligned inference it achieves the best Quality and dynamic degree overall (85.22 and 67.97), although its imaging quality is the lowest in the table (69.84). Mix inference outperforms Fixed250 in Quality

Table 9: **Full component and K/V bank ablations on 20-second Interactive generation** (expanding Table 2). Columns are grouped by inference policy: SAF^{M} (multiple stage-aligned banks), SAF^{S} (one Mix bank), and Fixed$t$ (one bank written at noise level t). In the first two groups, the w/o columns are training ablations evaluated with the same inference policy. ViCLIP averages the scores of the two 10-second segments. Best per group in **bold**, second underlined.

Metric	SAF^{M} (Aligned)			SAF^{S} (Mix)			Fixed$t$		
	Full	w/o K/V grad.	w/o sink recache	Full	w/o K/V grad.	w/o sink recache	t=750	t=500	t=250
Quality	**85.18**	84.64	81.70	**84.22**	84.02	81.30	**84.68**	84.12	83.99
ViCLIP	**26.54**	26.15	24.42	25.99	**26.01**	24.21	**25.71**	25.40	25.65
Subject Consistency	97.58	98.07	**99.24**	98.04	98.10	**99.10**	98.04	**98.25**	97.99
Background Consistency	96.56	97.01	**98.04**	96.92	97.04	**97.73**	96.90	**97.00**	96.90
Temporal Flickering	99.26	**99.38**	99.33	99.28	99.23	**99.29**	**99.24**	99.16	99.13
Motion Smoothness	98.93	99.07	**99.48**	99.06	99.07	**99.41**	98.94	99.02	**99.10**
Dynamic Degree	**64.10**	51.30	6.90	**48.00**	44.50	4.30	**55.60**	47.10	45.70
Aesthetic Quality	62.10	62.23	**62.58**	61.99	61.80	**62.45**	**62.02**	61.99	61.83
Imaging Quality	71.82	**72.58**	71.24	72.23	**72.72**	71.04	71.93	72.13	**72.36**
ViCLIP Segment 1	**27.06**	26.69	26.59	**27.10**	26.77	26.61	26.44	26.49	**27.14**
ViCLIP Segment 2	**26.01**	25.61	22.25	24.88	**25.24**	21.81	**24.97**	24.31	24.16

Table 10: **Full results across SDF training schedules on 20-second MovieGen-100s generation** (expanding Figure 4). Training schedules give the random/fixed/aligned history probabilities ($0/0/100$ is SAF), and each model is evaluated with Fixed250, Mix (SAF^{S}), and Aligned (SAF^{M}) inference. The columns after Quality are the 7 VBench-Long dimensions. Within each training schedule, best and second-best entries are **bold** and underlined.

Training	Inference	Quality	Subject	Background	Flicker	Smooth	Dynamic	Aesthetic	Imaging
0/0/100 (SAF)	Fixed250	83.97	**98.12**	**96.87**	98.64	**98.59**	50.39	63.98	70.71
	Mix	84.59	98.01	96.85	**98.82**	98.47	57.73	64.27	**70.87**
	Aligned	**85.22**	98.06	96.82	98.81	98.31	**67.97**	**64.84**	69.84
0/100/0	Fixed250	83.34	97.11	96.20	97.43	**98.76**	53.20	62.29	71.70
	Mix	82.57	**98.15**	**96.99**	98.05	98.68	31.48	63.92	**72.21**
	Aligned	**84.81**	98.09	96.95	**98.29**	98.26	**62.27**	**64.76**	71.41
100/0/0	Fixed250	83.40	98.34	97.12	98.67	**98.99**	36.09	64.81	71.29
	Mix	83.73	98.36	97.15	98.72	98.94	39.14	65.15	**71.57**
	Aligned	**84.67**	**98.50**	**97.37**	**98.78**	98.84	**50.08**	**65.64**	71.40
60/20/20	Fixed250	82.84	**98.85**	**97.73**	**99.30**	**99.32**	18.05	65.61	71.60
	Mix	82.99	**98.85**	97.69	99.24	99.26	20.39	**65.73**	**71.71**
	Aligned	**83.97**	98.55	97.35	98.89	98.92	**40.55**	65.51	71.15
30/35/35	Fixed250	83.50	**98.61**	**97.38**	**99.09**	**99.11**	30.39	**65.31**	72.02
	Mix	83.53	98.57	97.35	99.01	99.02	31.95	65.24	**72.18**
	Aligned	**84.74**	98.30	97.02	98.61	98.53	**55.86**	65.11	71.75

under every schedule except the fixed-only $0/100/0$. This model always reads history at t=250 during training, so Fixed250 is the inference policy most consistent with its training, whereas Mix also sharply reduces its dynamic degree, from 53.20 to 31.48. With Fixed250 or Mix inference, the $60/20/20$ schedule has the highest subject consistency, background consistency, temporal flickering, and motion smoothness, but the lowest dynamic degree. This again reflects the consistency–motion trade-off noted in Appendix D.

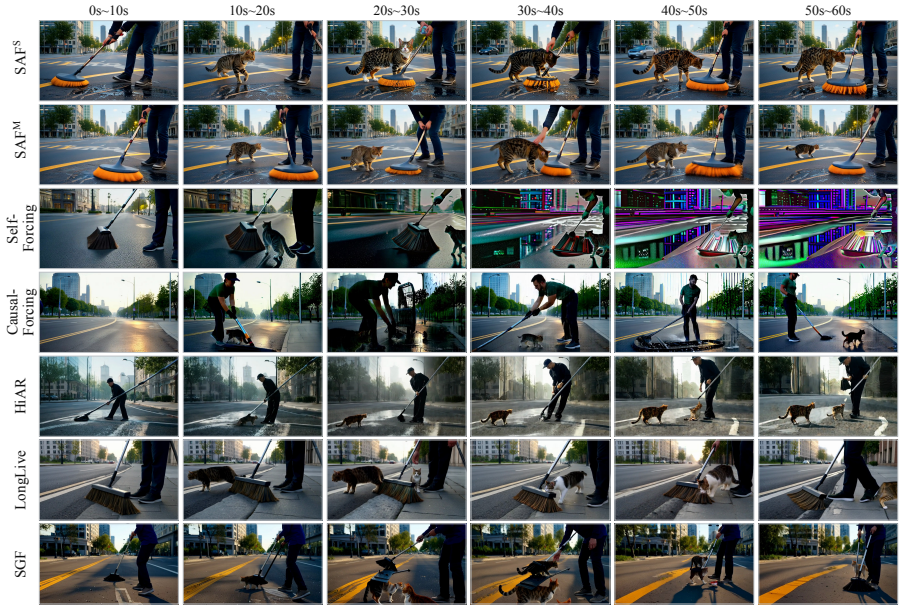
F Additional Qualitative Results

We show additional comparisons on Interactive and MovieGen-100s. The chunkwise figures (Figures 8–10) compare SAF^{S} and SAF^{M} with all five baselines. The framewise figures (Figures 11–13) compare them with Causal-Forcing and SGF, the two baselines that release framewise checkpoints. Within each figure, all methods use the same prompt or prompt sequence, the same sample seed, and the inference settings of the quantitative evaluation. Method names are shown on the left of each row. Interactive figures show the middle frame from each 10-second prompt interval and list the six prompts below the grid. MovieGen-100s figures show frames at 0, 20, 40, 60, 80, and 100 seconds. More visualizations are provided in the supplementary material.

G Limitations

Multiple versus single K/V banks. On a single GPU, SAF^{M} needs more memory than SAF^{S} because it stores one K/V bank per denoising stage. With pipelined inference (Algorithm 2), however, each GPU holds only the bank of its own stage, so the per-GPU memory is close to that of SAF^{S}. SAF^{S} saves memory by letting all stages read the same bank, which departs from the stage-aligned history used in training and allows errors to accumulate over long rollouts. The effect is minor in the chunkwise setting but more visible in the framewise setting, where single-frame blocks and a 21-frame window keep more of this history in context (Tables 1 and 5). We therefore recommend SAF^{M} with multi-GPU pipelined inference. SAF^{M} can also run on a single H100 GPU, so SAF^{S} is only needed when GPU memory is limited.

Inconsistent visual changes. We still observe inconsistent changes in two situations. When the camera moves, content newly entering at the frame boundary can change abruptly across frames, because the model can only refer to noisy K/V in the local window, and nothing outside the window except the sink is kept. At prompt switches, since SAF is trained only on single-prompt clips, changing the action is handled smoothly, but introducing new objects, scenes, or characters can make the appearance unstable, as the sink contains none of this content. Selectively retaining informative K/V beyond the window and training with prompt switches may address these two cases, which we leave to future work.



0-10 s: A street sweeper is methodically cleaning a city street with a push broom in the very early morning. The city is still asleep and the streets are empty. The streetlights are still on.

10-20 s: The street sweeper cleaning the empty city street is joined by a stray cat that cautiously approaches, meowing for attention. The streetlights cast long shadows.

20-30 s: The street sweeper and the stray cat are on the empty city street. The street sweeper stops his work, smiles, and takes out a small container of food from his cart to give to the cat.

30-40 s: The street sweeper on the empty city street watches as the stray cat eats the food hungrily. He gently pets the cat's back while it eats. The city is beginning to wake up.

40-50 s: The street sweeper and the stray cat on the city street. The cat, having finished its meal, rubs affectionately against the street sweeper's leg. The streetlights begin to turn off as the sun rises.

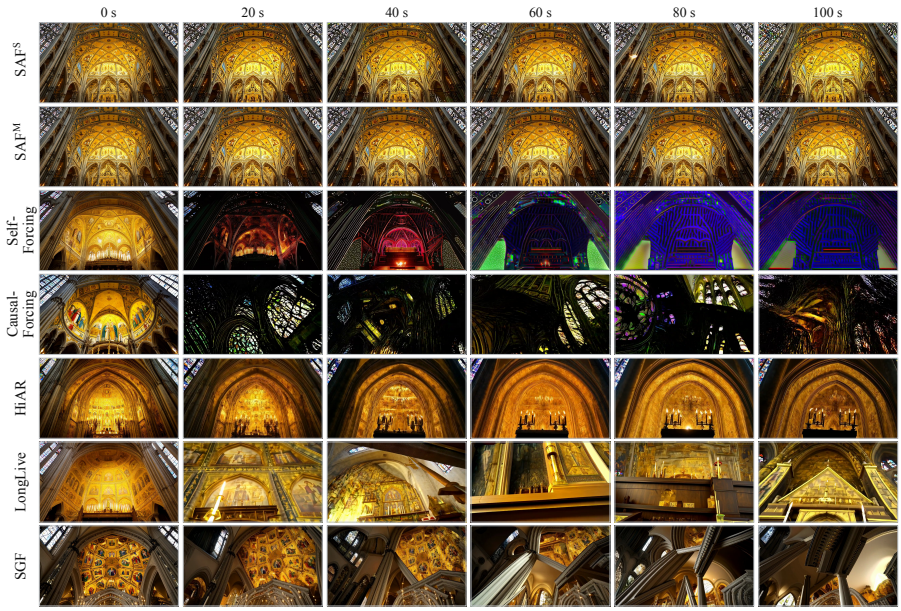
50-60 s: The street sweeper returns to his work, and the stray cat follows alongside him for a while, a temporary companion in the quiet, early morning hours on the city street.

Figure 8: **Additional chunkwise Interactive comparison.** Zoom in for better visualization.



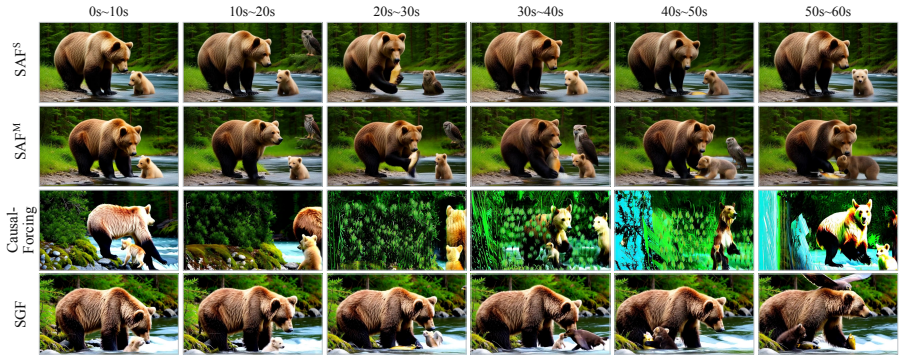
Prompt: A vintage-style photograph of a young woman in a flowing floral dress dropping a coin into a wishing well. She has wavy brown hair tied back with a ribbon, and her eyes sparkle with hope and determination as she gazes into the well. Her posture is upright, and her hand gently holds the coin before letting it drop. The background is a blurred scene of a quaint town square with old buildings and a few people walking by. The well itself is ornately carved with intricate designs, and the water ripples softly. The photo has a soft, nostalgic texture. A close-up shot from a slightly elevated angle.

Figure 9: **Additional chunkwise MovieGen-100s example 1.** Zoom in for better visualization.



Prompt: A dramatic tilt-down shot from the ceiling of a grand Gothic cathedral, revealing the intricate golden mosaics depicting biblical scenes and saints, with each tile meticulously arranged to form detailed patterns. The central focus is on the ornate altar below, adorned with candles and religious artifacts, creating a sacred and awe-inspiring atmosphere. The background features the soaring arches and stained glass windows, allowing a shaft of light to filter through, casting colorful hues across the mosaic floor. The scene has a detailed and realistic style, capturing the grandeur and solemnity of the cathedral interior.

Figure 10: **Additional chunkwise MovieGen-100s example 2.** Zoom in for better visualization.



0-10 s: A mother bear is standing at the edge of a rushing river, demonstrating to her small, fluffy cub how to catch fish. The river is in a dense pine forest. The cub watches intently.

10-20 s: The mother bear at the river's edge is observed by a wise old owl with large, yellow eyes, perched on a high branch of a nearby pine tree. The small bear cub is still watching its mother.

20-30 s: The mother bear and the wise old owl are at the river in the pine forest. The mother bear successfully swipes a salmon from the water. The owl hoots softly, as if in approval. The cub jumps up and down with excitement.

30-40 s: The mother bear, watched by the wise old owl, nudges the salmon towards her cub at the river's edge. The cub cautiously approaches the flapping fish. The pine forest is quiet except for the river.

40-50 s: The mother bear and the wise old owl in the pine forest watch as the cub playfully paws at the salmon, learning its first lesson in hunting. The mother bear looks on with pride.

50-60 s: The mother bear and her cub are eating the salmon by the river, while the wise old owl looks on from its perch in the pine tree before silently spreading its wings and flying deeper into the forest.

Figure 11: **Additional framewise Interactive comparison.** Zoom in for better visualization.



Prompt: A vibrant and lively scene from a colorful Indian festival in Mumbai, where a toy robot wearing blue jeans and a white T-shirt takes a pleasant stroll. The robot has a friendly expression, with its arms swinging naturally as it walks along the bustling streets. The background is filled with people in traditional attire, vibrant decorations, and colorful lights, creating a festive atmosphere. The festival is alive with music and dance, and there are stalls selling various sweets and snacks. The robot appears to be enjoying the festivities, with its legs moving in a casual, relaxed manner. The camera angle is slightly elevated, capturing both the robot and the vibrant surroundings.

Figure 12: **Additional framewise MovieGen-100s example 1.** Zoom in for better visualization.



Prompt: A high-speed video capturing the moment champagne is poured into a glass, with bubbles rising rapidly and cascading down the sides. The glass is clear and elegant, reflecting the sparkling liquid inside. The bubbles form and pop with each other, creating a lively and dynamic scene. The background is a blurred, dimly lit room, emphasizing the focus on the champagne. The camera angle is from below, providing a dramatic perspective of the pouring action.

Figure 13: **Additional framewise MovieGen-100s example 2.** Zoom in for better visualization.