MS. FORCING: EFFICIENT STREAMING VIDEO GENERATION WITH MULTI-SCALE PATCHIFICATION AND ATTENTION

Anonymous authors
Paper under double-blind review

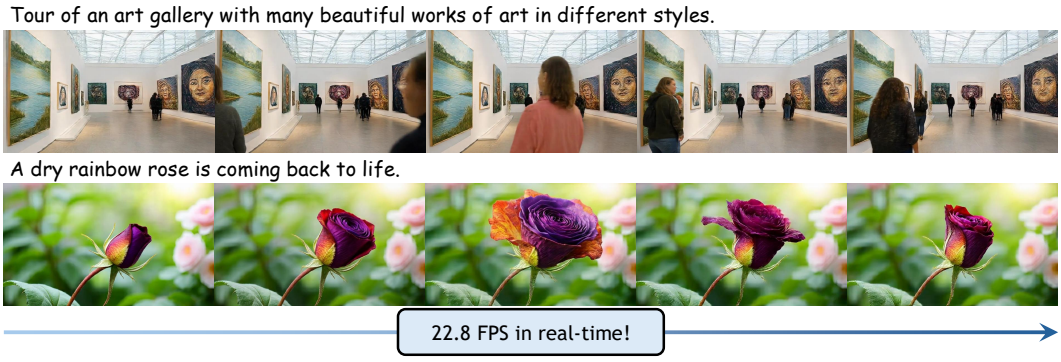

Figure 1: **Ms. Forcing** is a real-time streaming video generation framework (22.8 FPS) built on rolling diffusion. It combines our novel Multi-Scale Patchification (MSP) and Multi-Scale Self-Attention (MSSA), which adapt the spatial granularity of computation to each state's noise level in the rolling denoising window, with Homogeneous-Noise-Level DMD (H-DMD), which reduces the mismatch between DMD training sequences and inference-time rollouts.

ABSTRACT

Streaming video diffusion models have made substantial progress toward interactive and dynamic world simulation, but the nested autoregressive and denoising loops of conventional next-frame generation hinder real-time deployment. Recent rolling-window methods pipeline denoising across multiple consecutive frames at different noise levels, improving throughput and long-horizon stability. However, they tokenize every state at the same fine spatial granularity, leaving substantial noise-dependent redundancy in the joint denoising window. We propose **Ms. Forcing**, which adapts spatial granularity to each state's noise level for efficient streaming video generation. Its *Multi-Scale Patchification* (MSP) assigns coarser patches to noisier states, reducing the active-window token count by 45%, while *Multi-Scale Self-Attention* (MSSA) matches the density of visible non-sink keys and values to each query scale to further reduce attention cost. Fixing both schedules by window position preserves a static, hardware-friendly computation graph in Ms. Forcing. We further reduce the training–inference mismatch in rolling-window DMD through *Homogeneous-Noise-Level DMD* (H-DMD), which assembles each fake video from clean predictions sharing the same source noise level. The multi-scale design helps offset the additional training cost of backpropagating through overlapping windows. We include both quantitative and qualitative experiments to show that Ms. Forcing reaches **22.84 FPS on a single H200 GPU**, 39.7% faster than Rolling Forcing, while significantly improving VBench scores in both short video and long video generation settings.

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

1 INTRODUCTION

Large-scale video diffusion models (Wan et al., 2025; HaCohen et al., 2026; Team et al., 2025) generate realistic videos with intricate dynamics and high visual quality. However, synthesizing a complete clip through iterative bidirectional denoising incurs high latency and precludes on-demand output. Interactive applications such as generative world models (Bruce et al., 2024; Gao et al., 2026) and real-time virtual avatars (Sun et al., 2026; Huang et al., 2026) instead require *streaming* generation: frames must be emitted sequentially with low latency while remaining coherent over long horizons. Delivering real-time throughput without sacrificing visual fidelity, motion quality, or long-horizon stability remains a central challenge.

Recent methods enable streaming by distilling slow bidirectional diffusion models into few-step causal autoregressive generators (Yin et al., 2025; Huang et al., 2025). A conventional causal sampler nests an inner denoising loop for each new frame inside an outer autoregressive loop. Each frame is finalized before the next is generated, so errors propagate along the autoregressive trajectory. Also, inference efficiency constraints will limit the history context length and further exacerbate long-horizon drift (Huang et al., 2025). Liu et al. (2026) instead introduce Rolling Forcing (RF), which pipelines T denoising steps along the temporal axis through a joint window of T consecutive frames at progressively increasing noise levels. Each pass advances all window states by one noise level and emits the clean head frame. Bidirectional attention within the window lets frames refine one another before emission, expanding the denoising context and reducing error accumulation while preserving causal output.

RF trades nested sampling for a heavier forward pass: all T active states are tokenized at the backbone's native patch size despite their different noise levels. Pruning or compressing history KVs (Guo et al., 2026; Yang et al., 2026) reduces the cost of attending to past context, but leaves the token count within the joint denoising window unchanged. To reduce this token count, we draw on the coarse-to-fine nature of diffusion denoising: high-noise states primarily recover low-frequency structure, whereas low-noise states refine high-frequency details (Kim et al., 2026; Jin et al., 2025). Because each window position in RF corresponds to a prescribed noise level, we can match spatial granularity to noise level through a predefined schedule, without input-dependent routing. Motivated by the zero-shot probes of redundancy in the joint denoising window in Section 3.2, we propose *Multi-Scale Patchification* (MSP), which assigns coarser patchification to noisier states, and *Multi-Scale Self-Attention* (MSSA), which matches the density of visible non-sink KVs to each query's patch granularity while preserving the full-resolution sink frame. Under our 5-step RF schedule, MSP reduces tokens in the denoising window by 45%, and MSSA further removes 12.9% of post-MSP dot-product attention computation. Both are fixed by window position, yielding a static compute graph.

The reduced training cost also enables a more faithful construction for Distribution Matching Distillation (DMD; Yin et al., 2024a;b). To control memory, RF builds its fake video from non-overlapping windows by concatenating all T predicted clean states from each selected window. Adjacent frames therefore originate from different rollout noise levels, whereas inference concatenates head predictions from consecutive windows at a common source noise level. DMD thus supervises fake videos whose frame-to-frame source-noise pattern differs from that of inference-time outputs. We propose *Homogeneous-noise-level DMD* (H-DMD) for joint denoising: at each update, it samples one window position and assembles a fake video from that position's predicted clean state in every consecutive window. All frames in the assembled fake video then share one source noise level, avoiding frame-to-frame variation in source noise level without changing the DMD objective. Although it requires gradients through overlapping windows, MSP and MSSA make it tractable and remove the need for auxiliary Self-Forcing (Huang et al., 2025) rollout updates.

Together, these designs form **Ms. Forcing**. At 832×480 resolution, it reaches 22.84 FPS on a single H200 GPU (39.7% faster than RF), improves five-second VBench quality and semantic scores over RF by 0.29 and 1.25, and reduces 60-second quality drift from 2.227 to 1.700 while maintaining competitive quality. Our contributions are:

- We introduce a static noise-matched schedule through **Multi-Scale Patchification** and **Multi-Scale Self-Attention**, reducing tokens in the denoising window by 45% and post-MSP attention computation by a further 12.9%.

- We identify a training–inference mismatch introduced by RF's heterogeneous DMD assembly and propose **Homogeneous-Noise-Level DMD**, which improves distillation effectiveness and generation quality without requiring auxiliary Self-Forcing training.

2 Related Work

Autoregressive Video Diffusion Models. Large-scale video diffusion models (VDMs) (Wan et al., 2025; HaCohen et al., 2026; Team et al., 2025; Yang et al., 2025; Kong et al., 2024; Seedance et al., 2026; Cong et al., 2026; Du et al., 2026; Li et al., 2026) typically synthesize entire clips using bidirectional attention, making them unsuitable for low-latency streaming. To transform these slow bidirectional VDMs into efficient streaming generators, recent work combines autoregressive modeling with iterative diffusion denoising under several formulations (Chen et al., 2024). Most existing streaming video generators (Yin et al., 2025; Huang et al., 2025; Lu et al., 2026) adopt a nested sampling process: an outer autoregressive loop generates frames sequentially, while an inner diffusion loop progressively denoises each new frame. A complementary line of work (Ruhe et al., 2024; Kim et al., 2024; Sun et al., 2025; Liu et al., 2026) pipelines the denoising steps along the temporal axis, jointly updating consecutive blocks at progressively increasing noise levels within a rolling window and emitting one clean block per forward pass. To mitigate exposure bias and improve generation quality, recent approaches further apply distribution matching distillation (DMD) to self-generated rollouts, thereby better aligning training with the inference-time distribution (Huang et al., 2025).

Efficient Video Diffusion. Beyond reducing the number of denoising steps through distillation, complementary acceleration methods eliminate redundant computation or lower the cost of each model evaluation. For efficient and effective streaming inference, sliding-window attention and persistent sink frames bound the history context length while retaining critical long-range context (Yang et al., 2026; Shin et al., 2026). Recent methods further reduce history cost through KV-cache token pruning or compression (Guo et al., 2026; Chen et al., 2026; Yi et al., 2026). Other acceleration strategies reuse intermediate features across adjacent denoising steps (Liu et al., 2025b;a), perform staged coarse-to-fine generation (Jin et al., 2025; Ding et al., 2026; Zheng et al., 2026), or use adaptive patch schedules during denoising (Kim et al., 2026). Most of them rely on runtime importance estimation or adaptive routing, which can introduce input-dependent and irregular execution. More importantly, acceleration within the joint denoising window (Ruhe et al., 2024; Liu et al., 2026) remains largely unexplored. We exploit the deterministic correspondence between window position and noise level to assign noise-matched patch sizes and history receptive fields a priori, yielding a static, hardware-friendly schedule.

3 Preliminary and Motivation

3.1 Preliminary: Autoregressive video diffusion models

Autoregressive video diffusion factorizes the distribution of an N-frame sequence $\boldsymbol{x}^{1:N}$ as $p(\boldsymbol{x}^{1:N}) = \prod_{i=1}^{N} p(\boldsymbol{x}^i \mid \boldsymbol{x}^{<i})$, where each $\boldsymbol{x}^i$ may represent a short frame chunk, referred to as a "frame" for simplicity. For efficient streaming computation, recent works (Lu et al., 2026; Yang et al., 2026) organize context tokens as $[\,\text{sink} \mid \text{recent} \mid \text{now}\,]$ under causal attention and generate each new block through few-step denoising (Yin et al., 2025; Huang et al., 2025; Yang et al., 2026). Given a T-step diffusion ($t_1 \le \cdots \le t_T$) model p_θ, the generation of frame i can be formulated as

$$p(\boldsymbol{x}^i_{t_{0:T}} \mid \boldsymbol{x}^{<i}) = \prod_{j=1}^{T} p_\theta(\boldsymbol{x}^i_{t_{j-1}} \mid \boldsymbol{x}^i_{t_j}, \boldsymbol{x}^{<i}), \qquad \boldsymbol{x}^i_{t_T} \sim \mathcal{N}(0, \mathbf{I}). \tag{1}$$

While they enable consistent sequential generation, their strictly causal frame prediction causes each frame to inherit errors from its predecessors, allowing small imperfections to compound over long horizons and eventually leading to noticeable drift and quality degradation. To this end, Rolling Forcing (RF) pipelines the T denoising steps of each generated frame along the temporal axis via a joint rolling window containing T consecutive frames assigned progressively higher noise levels $t_1 \le \cdots \le t_T$. Rather than fully denoising each frame before moving to the next, RF expands the denoising context length and allows frames to refine one another through bidirectional attention

before finalization, reducing local errors and long-horizon drift while preserving causal streaming. At each rolling step, RF jointly denoises frames in the window by one noise level in a single forward pass, sampling from the transition

$$p_\theta\left(\boldsymbol{x}_{t_{0:T-1}}^{i:i+T-1} \,\middle|\, \boldsymbol{x}_{t_{1:T}}^{i:i+T-1},\, \boldsymbol{x}^{<i}\right), \qquad \boldsymbol{x}_{t_T}^{i+T-1} \sim \mathcal{N}(0, \mathbf{I}). \tag{2}$$

Position l contains $\boldsymbol{x}_{t_l}^{i+l-1}$, ordered from the near-clean head at t_1 to the near-noise tail at t_T. After each pass, RF emits the denoised head frame, advances the remaining frames toward t_0, and appends a fresh noise frame $\boldsymbol{x}_{t_T}^{i+T} \sim \mathcal{N}(0, \mathbf{I})$ at the tail, allowing the window to roll forward indefinitely.

3.2 Zero-Shot Probes of Redundancy in the Joint Denoising Window

Diffusion denoising proceeds from coarse to fine: denoising at high noise levels primarily establishes global structure, whereas low-noise steps refine local details. This observation has motivated prior diffusion models to process high-noise states at lower spatial resolutions (Kim et al., 2026; Jin et al., 2025). Unlike conventional denoising inference, which processes inputs at a single noise level during each denoising pass, RF jointly processes a mixed-noise-level input within the bidirectional denoising window. Yet RF patchifies all frames in this joint denoising window at a uniform spatial granularity. We therefore test whether this mixed-noise denoising window can instead be represented as a mixed-granularity input sequence based on the noise schedule. To this end, we adopt two perturbations during zero-shot pretrained RF rollouts, probing the sensitivity of the hybrid granularity of patchification and self-attention via frame-wise LPIPS (Zhang et al., 2018).

Probing patchification granularity. Direct zero-shot mixed-granularity patchification is infeasible because each new patch size requires learned patch-embedding and de-embedding layers. We therefore use a prediction-space proxy for coarse granularity: at the three highest source noise levels, we apply non-overlapping 2×2 average pooling to each predicted clean latent $\hat{\boldsymbol{x}}_0$, followed by nearest-neighbor upsampling to its original spatial resolution. Under this proxy, the LPIPS deviation from the paired baseline remains below 0.26 throughout a rollout of over 10 seconds (blue curve in Fig. 2), motivating mixed-granularity patchification within joint denoising windows.

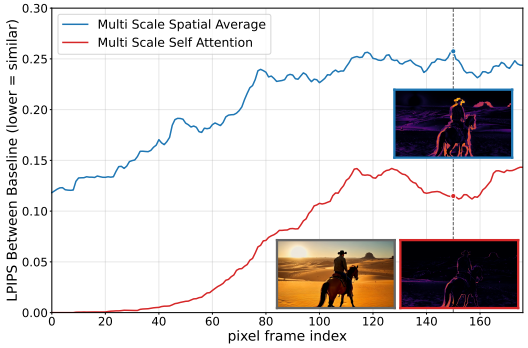

Figure 2: **Zero-shot probes of mixed granularity.** Frame-wise LPIPS (↓) between each perturbed RF and the original rollout across 177 pixel frames. For 5-step RF, we treat the three highest-noise levels as coarse. Blue: 2×2 average pooling of predicted clean latents at these levels, followed by nearest-neighbor upsampling; Red: For *queries* at these levels, we subsample non-sink history KVs with stride 2, preserving the sink and current window.

Probing attention-context granularity. For selected noise levels, we spatially subsample non-sink history *keys* and *values* based on the stride assigned to each noise level, preserving the sink and current window. This reduces the spatial density of KV-context for high-noise *queries*. The LPIPS remains below 0.15 throughout the rollout (red curve in Fig. 2), showing stable rollout behavior under the resulting mixed-granularity self-attention of the joint denoising window.

4 Methodology

4.1 Multi-Scale Patchification and Self-Attention

In this section, we introduce the *Multi-Scale Patchification* (MSP) and *Multi-Scale Self-Attention* (MSSA), which are built upon the RF joint denoising schema. As illustrated in Fig. 3, MSP assigns a noise-matched mixed-granularity patch setting within the window, while MSSA matches the spatial density of its attention-visible keys and values to that granularity. The resulting scale tiers are processed jointly by a shared DiT with lightweight tier-specific modules, without dynamic routing.

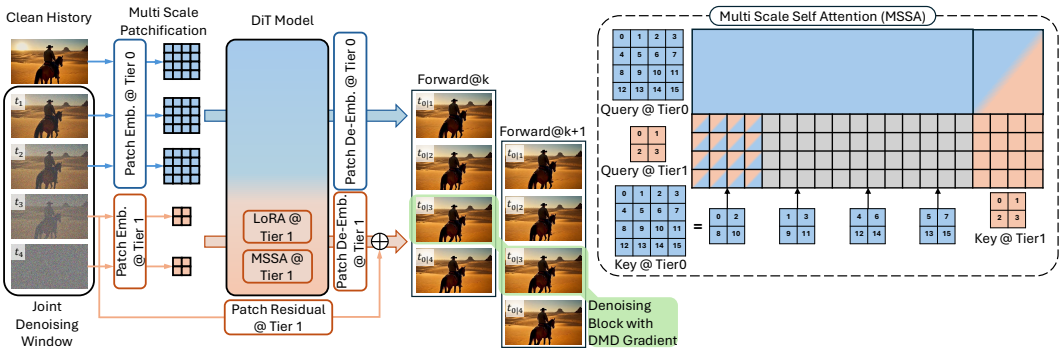

Figure 3: **Overview of Ms. Forcing.** Multi-Scale Patchification (MSP) maps the mixed noise levels in a joint denoising window to static patch-size tiers, which are processed by a shared DiT with lightweight tier-specific modules. Multi-Scale Self-Attention (MSSA) matches the spatial density of attention-visible keys and values to each query tier without changing the attention mask. As the window rolls, each state is re-tokenized from coarse to fine; predicted clean states at a shared noise level form the homogeneous sequence used for DMD training.

Multi-Scale Patchification. In the joint denoising window at *forward@k*, latent frame $\boldsymbol{x}_{t_l}^{k+l-1}$ at position l is assigned a patch size p_l under the monotonic schedule $p_1 \le \cdots \le p_T$, $p_l \in \{p, 2p, 4p, \cdots\}$, as shown in Fig. 3. Each patch size defines a tier with a complete scale-specific processing module, comprising patch embedding and de-embedding layers, LoRA adapters, and a latent residual branch. The native p-tier directly reuses the pretrained model's original pathway and parameters, whereas each coarser tier has its own separately parameterized module; the underlying DiT backbone remains shared across tiers. To quantify the saving, we compare against RF, which tokenizes all T window states at the native patch size and therefore uses $N_{\mathrm{RF}} = THW/p^2$ tokens per window. For a latent frame with spatial size $H \times W$, MSP instead uses

$$N_l^{\mathrm{MSP}} = \frac{HW}{p_l^2}, \qquad N_{\mathrm{MSP}} = \sum_{l=1}^{T} N_l^{\mathrm{MSP}}, \qquad \rho_{\mathrm{tok}} \triangleq \frac{N_{\mathrm{MSP}}}{N_{\mathrm{RF}}} = \frac{1}{T}\sum_{l=1}^{T}\left(\frac{p}{p_l}\right)^2. \tag{3}$$

Here, ρ_{tok} is the fraction of tokens retained by MSP in the joint denoising window relative to RF. For the 5-step RF with $(p, p, 2p, 2p, 2p)$ used in our experiments, $\rho_{\mathrm{tok}} = 0.55$, corresponding to a 45% token reduction in the joint denoising window.

Multi-Scale Self-Attention. Motivated by the attention-context granularity probe in Sec. 3.2, MSSA removes the remaining attention redundancy by making each visible non-sink KV context no denser than the query representation, while leaving the attention-sink KVs unchanged (Fig. 3). After MSP, let $\mathcal{C}_{\mathrm{sink}}$ and $\mathcal{C}_b$ denote the KV contexts from the attention sink and the non-sink frame at position b, respectively, and let $\mathcal{V}(a)$ be the non-sink frame positions visible to queries at position a under the original attention mask. For the query tokens at position a, MSSA constructs the attended KV context as

$$\mathcal{C}_a^{\mathrm{MSSA}} = \left[\mathcal{C}_{\mathrm{sink}} \,\middle\|\, \underset{b\in\mathcal{V}(a)}{\mathrm{Concat}}\,\mathrm{Subsample}\left(\mathcal{C}_b,\ \mathrm{stride} = \max\left(1, \frac{p_a}{p_b}\right)\right)\right], \tag{4}$$

To quantify the incremental gain over MSP, define the normalized token fraction $\eta_l = (p/p_l)^2$ and the two-dimensional spatial stride $s_{ab} = \max(1, p_a/p_b)$. The resulting reduction in dot-product attention computation is

$$r_{\mathrm{MSSA}} = 1 - \left[\sum_{a=1}^{T}\eta_a\left(1 + \sum_{b\in\mathcal{V}(a)}\eta_b/s_{ab}^2\right) \Big/ \sum_{a=1}^{T}\eta_a\left(1 + \sum_{b\in\mathcal{V}(a)}\eta_b\right)\right]. \tag{5}$$

For the 5-step RF layout with one sink frame, one recent-history frame, and five frames in the joint denoising window, substituting the MSP schedule (p,p,2p,2p,2p) into Eq. 5 yields $r_{\mathrm{MSSA}} \approx 12.9\%$.

Algorithm 1 Ms. Forcing Training with Homogeneous-noise-level DMD

Require: Rollout levels $\{\tau_0, \tau_1, \ldots, \tau_T\}$; patch sizes $p_1 \le \cdots \le p_T$; sequence length N
Require: Generator G_θ; fake-score network s_ϕ; frozen teacher score s_{teacher}
loop
  Initialize $\mathbf{X}_\theta \leftarrow [\,]$, $\mathbf{KV} \leftarrow [\,]$, and warm up the first $T-1$ rolling states
  Sample $j \sim \text{Uniform}\{1, \ldots, T\}$ ▷ shared rollout level
  for $i = 1, \ldots, N$ **do**
    Append $\boldsymbol{x}_{\tau_T}^{i+T-1} \sim \mathcal{N}(0, \mathbf{I})$ to form $\boldsymbol{x}_{\tau_{1:T}}^{i:i+T-1}$
    $\hat{\mathbf{W}}_0^{(i)} \leftarrow G_\theta(\boldsymbol{x}_{\tau_{1:T}}^{i:i+T-1}, \tau_{1:T}, \mathbf{KV})$ ▷ MSP and MSSA
    $\mathbf{X}_\theta.\texttt{append}\big(\pi_j(\hat{\mathbf{W}}_0^{(i)})\big)$ ▷ Eq. 8
    Update $\mathbf{KV}$ with $\text{sg}[\pi_1(\hat{\mathbf{W}}_0^{(i)})]$
    $\boldsymbol{x}_{\tau_{1:T-1}}^{i+1:i+T-1} \leftarrow \Psi\Big(\text{sg}[\pi_{2:T}(\hat{\mathbf{W}}_0^{(i)})], \tau_{1:T-1}\Big)$
  end for
  Update ϕ with the diffusion denoising loss on $\text{sg}[\mathbf{X}_\theta]$
  if generator-update step **then**
    Sample DMD timestep t and noise $\boldsymbol{\epsilon}$; set $\mathbf{X}_t = \alpha_t \mathbf{X}_\theta + \sigma_t \boldsymbol{\epsilon}$
    Freeze s_ϕ and update θ with Eq. equation 6 on $\mathbf{X}_t$
  end if
end loop

4.2 Homogeneous-noise-level assembly for DMD

Few-step streaming generators are commonly trained with distribution matching distillation (DMD; Yin et al., 2024b;a) on self-generated rollouts (Huang et al., 2025; Liu et al., 2026) to minimize the reverse KL divergence between real $p_{\text{real}}(\boldsymbol{x})$ and generated distributions $p_{\text{fake}}(\boldsymbol{x})$ across timesteps while reducing the training-inference gap in autoregressive generation:

$$\begin{aligned}\nabla_\theta \mathcal{L}_{\text{DMD}} &\triangleq \mathbb{E}_t(\nabla_\theta \mathbb{D}_{\text{KL}}(p_{\text{fake},t}(\boldsymbol{x}_t) || p_{\text{real},t}(\boldsymbol{x}_t))) \\ &\approx -\mathbb{E}_t\Big(\int (s_{\text{real}}(\Psi(G_\theta(\epsilon), t), t) - s_{\text{fake}}(\Psi(G_\theta(\epsilon), t), t)) \frac{\mathrm{d}G_\theta(\epsilon)}{\mathrm{d}\theta} \mathrm{d}\epsilon\Big),\end{aligned} \tag{6}$$

where $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and Ψ denotes forward diffusion at timestep t. In diffusion models, the score function is defined as: $s_{\text{real}}(\boldsymbol{x}_t, t) = \nabla_{\boldsymbol{x}_t} \log p_{\text{real},t}(\boldsymbol{x}_t) = -\frac{\boldsymbol{x}_t - \alpha_t \mu_{\text{real}}(\boldsymbol{x}_t, t)}{\sigma_t^2}$, where μ_{real} is the denoised estimate, and α_t, σ_t are noise schedule parameters (Ho et al., 2020; Nichol & Dhariwal, 2021; Karras et al., 2022). DMD freezes pretrained μ_{real} (teacher) and updates μ_{fake} on generator outputs.

Rolling Forcing samples a subset of non-overlapping windows to construct the predicted clean video for DMD training where gradient computation is performed only on these selected windows, significantly reducing memory usage while retaining effective supervision.

$$\hat{\boldsymbol{x}}_{0,\text{RF}}^{1:N} = \underset{\substack{1 \le i \le N \\ i \equiv r \pmod T}}{\text{Concat}} G_\theta\big(\boldsymbol{x}_{t_{1:T}}^{i:i+T-1}, t_{1:T}, \boldsymbol{x}_0^{<i}\big), \qquad r \sim \text{Uniform}\{0, \ldots, T-1\}. \tag{7}$$

However, Eq. 7 concatenates clean predictions from different rollout noise levels $t_{1:T}$, producing a heterogeneous fake sequence with level-dependent temporal marginals. RF mitigates this issue by equally alternating Self-Forcing and Rolling-Forcing updates, but the latter still exposes DMD to variations in clarity and motion that may induce unnatural appearance changes or camera movement.

Instead, at each iteration, we propose H-DMD and sample a window position j and collect the corresponding clean prediction from every consecutive rolling window:

$$\hat{\boldsymbol{x}}_{0,\text{Our}}^{1:N} = \underset{i=1}{\overset{N}{\text{Concat}}} \pi_j\big(G_\theta\big(\boldsymbol{x}_{t_{1:T}}^{i:i+T-1}, t_{1:T}, \boldsymbol{x}_0^{<i}\big)\big), \qquad j \sim \text{Uniform}\{1, \ldots, T\}. \tag{8}$$

Here π_j selects the clean prediction associated with noise level t_j in each window. Consequently, every assembled clean predicted video is homogeneous in its source noise level. By reducing token and attention computation, MSP and MSSA keep training tractable despite backpropagating the DMD gradient through all overlapping joint-denoising windows. The complete training procedure is summarized in Algorithm 1.

5 EXPERIMENTS

5.1 IMPLEMENTATION DETAILS

Model. We implement Ms. Forcing with Wan2.1-T2V-1.3B (Wan et al., 2025) as our base model, which generates 16 FPS videos with a resolution of 832×480. Following Self Forcing (Huang et al., 2025) and Rolling Forcing (Liu et al., 2026), we first initialize the base model with causal attention masking on 16k ODE solution pairs sampled from the base model.[1] Then we use another 6K ODE solution pairs from Causal Forcing (Zhu et al., 2026) to warm up the patch embedding, patch de-embedding, patch residual, and LoRA, using AdamW with a batch size of 8 and initial learning rate 10^{-4} for 1000 steps. We set $T = 5$ and each chunk in the joint denoising window contains 3 latent frames. In DMD stage, the model is trained for 1400 steps with a batch size of 64 and a trained temporal window of 21 latent frames. We use the AdamW optimizer for both the generator G_θ (learning rate 1.5×10^{-6}) and the fake score s_{gen} (learning rate 4.0×10^{-7}). The generator is updated every 5 steps of fake score updates.

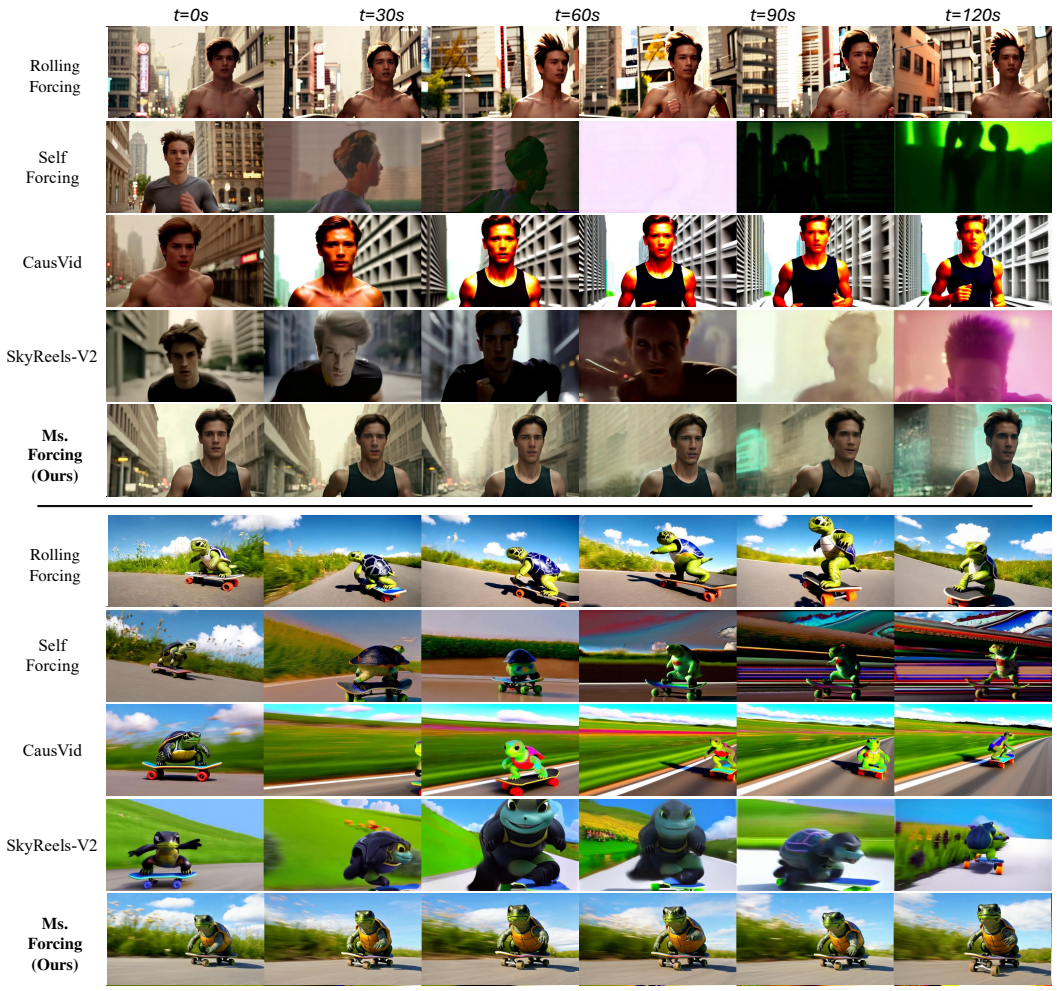

Figure 4: **Qualitative comparison on long-horizon generation.** Each row shows frames sampled from a long-horizon rollout, with labeled timestamps ranging from 0 to 120 s. Compared with the baselines, Ms. Forcing retains sharpness, consistent color tone, and prompt adherence in late frames, whereas other methods exhibit exposure drift and high-frequency artifacts.

[1] We use the official ODE init checkpoint provided by Self Forcing.

Evaluation. We evaluate short- and long-video generation using VBench-based metrics (Huang et al., 2024), excluding the VLM-based scores of Reward Forcing (Lu et al., 2026). For our short-video evaluation, we generate 5-second clips from the same 946 rewritten official VBench prompts, using five random seeds per prompt, and report the standard VBench total, quality, and semantic scores (Huang et al., 2024). For long videos, we use the first 128 MovieGen benchmark prompts (Polyak et al., 2024) to generate 60-second videos and evaluate subject consistency, background consistency, motion smoothness, dynamic degree, aesthetic quality, and imaging quality. The individual metrics are normalized and aggregated using the standard VBench coefficients. All videos are generated at 16 FPS and a resolution of 832×480. Following the same protocol, we quantify long-video quality drift as the standard deviation of imaging-quality scores over 30 non-overlapping 2-second segments of each video, averaged across the evaluation set; lower values indicate more stable visual quality. We report inference throughput in FPS on a single H200 GPU. Additional implementation details are provided in Appendix A.

5.2 Qualitative Results

Figure 4 compares Ms. Forcing with the baselines at timestamps from 0 to 120 seconds in long-horizon autoregressive rollouts. Ms. Forcing maintains high visual quality and semantic consistency even in later frames, whereas methods with sparser long-range context suffer from exposure drift and high-frequency artifacts. Additionally, Ms. Forcing achieves the fastest inference speed among the compared methods while incurring substantially lower training costs than Rolling Forcing.

5.3 Quantitative Comparisons and Ablations

We compare Ms. Forcing with representative streaming video generators of comparable scale. These include the Diffusion-Forcing-based SkyReels-V2 (Chen et al., 2025; 2024) and the distilled causal models CausVid (Yin et al., 2025), Self Forcing (Huang et al., 2025), LongLive (Yang et al., 2026), and Rolling Forcing (Liu et al., 2026). Rolling Forcing is our direct baseline, while the other methods represent complementary strategies for causal distillation and long-horizon generation.

Table 1: **Short-video comparison and cumulative ablation.** H-DMD, MSP, and MSSA are cumulatively added to Rolling Forcing. VBench scores for the first four methods are taken from published results; those for Rolling Forcing and its ablations are evaluated by us. Best and second-best results across all displayed rows are **bold** and underlined, respectively.

Model	Params	FPS↑	VBench evaluation scores ↑		
			Total	Quality	Semantic
SkyReels-V2 (Chen et al., 2025)	1.3B	0.43	82.67	**84.70**	74.53
CausVid (Yin et al., 2025)	1.3B	15.92	82.88	83.93	78.69
Self Forcing (Huang et al., 2025)	1.3B	15.92	83.80	84.59	80.64
LongLive (Yang et al., 2026)	1.3B	19.85	83.22	83.68	**81.37**
Component ablation (cumulative)					
Rolling Forcing (Liu et al., 2026)	1.3B	16.35	82.88	83.63	79.88
+ H-DMD	1.3B	16.35	83.10	83.57	81.22
+ MSP	1.3B	21.93	**83.89**	84.61	81.01
+ MSSA **(Ms. Forcing)**	1.3B	**22.84**	83.36	83.92	81.13

Short-video quality and efficiency. Table 1 reports the VBench results and cumulative component ablation on 5-second videos. Ms. Forcing achieves the highest throughput among all compared models, while its total and semantic scores are within 0.44 and 0.24 of the best external baselines, respectively. The ablation further disentangles the contribution of each component. H-DMD raises the semantic and total scores by 1.34 and 0.22, respectively, while maintaining throughput at 16.35 FPS, showing that homogeneous noise-level assembly improves text–video alignment without additional inference cost. The complete model with MSP and MSSA **increases throughput by** 39.7% **over RF to** 22.84 **FPS**, while improving quality and semantic scores over RF. The cumulative gains are not uniform: relative to H-DMD alone, the complete model improves quality but slightly lowers the semantic score; adding MSSA to MSP increases throughput and the semantic score at the cost

of lower total and quality scores. Together with the lower long-video drift discussed below, these results characterize the trade-off between efficiency and generation quality.

Table 2: **Long video generation comparison.** H-DMD, MSP, and MSSA are cumulatively added to Rolling Forcing. Best and second-best results are **bold** and underlined, respectively.

Model	VBench Long Evaluation Scores ↑							Drift↓
	Total	Subject	Background	Smoothness	Dynamic	Aesthetic	Imaging Quality	
Diffusion Forcing								
SkyReels-V2 (Chen et al., 2025)	75.94	96.43	96.59	**98.91**	39.86	50.76	58.65	7.315
Distilled Causal								
CausVid (Yin et al., 2025)	77.78	97.92	**96.62**	98.47	27.55	58.39	67.77	2.906
Self Forcing (Huang et al., 2025)	79.34	97.10	96.03	98.48	**54.94**	54.40	67.61	5.075
LongLive (Yang et al., 2026)	79.53	**97.96**	96.50	98.79	35.54	57.81	69.91	2.531
Component ablation (cumulative)								
Rolling Forcing (Liu et al., 2026)	79.38	97.78	96.41	98.73	33.67	60.06	**70.66**	2.227
+ H-DMD	**80.67**	97.51	95.98	98.67	51.80	**60.11**	69.72	3.041
+ MSP	79.95	97.50	96.07	98.56	48.78	59.06	68.60	2.582
+ MSSA (**Ms. Forcing**)	79.66	97.90	96.58	98.70	41.85	58.89	69.04	**1.700**

Long-video quality and drift. Table 2 evaluates 60-second generation, where error accumulation becomes more pronounced. H-DMD substantially increases the dynamic degree of Rolling Forcing from 33.67 to 51.80, indicating stronger motion dynamics over long-horizon rollouts, but this improvement comes with increased quality drift. Removing redundant fine-grained computation from the joint denoising window with MSP and MSSA progressively reduces the drift to 2.582 and 1.700, respectively. This stabilization is accompanied by a gradual decrease in dynamic degree, yet the complete Ms. Forcing model remains substantially more dynamic than Rolling Forcing while achieving markedly lower drift. Overall, Ms. Forcing strikes a favorable balance between motion dynamics and long-horizon stability while maintaining competitive generation quality across the remaining metrics and delivering faster streaming inference speed than Rolling Forcing.

6 Conclusion

With **Ms. Forcing**, we reduce the fixed-granularity computation of rolling-window denoising for efficient streaming video generation. Exploiting the deterministic correspondence between window position and noise level, *Multi-Scale Patchification* (MSP) assigns coarser patches to noisier states and reduces the number of tokens in the joint denoising window by 45%. *Multi-Scale Self-Attention* (MSSA) further reduces post-MSP dot-product attention computation by 12.9% by matching the density of the visible non-sink KV context to each query's patch granularity while preserving a full-resolution attention sink. Both schedules are fixed by window position and therefore remain static and hardware-friendly. We also introduced *Homogeneous-Noise-Level DMD* (H-DMD), which replaces RF's heterogeneous fake-video assembly with sequences whose predictions share a common source noise level, reducing the mismatch with inference-time rollouts without auxiliary Self-Forcing training. At 832×480 resolution, Ms. Forcing increases throughput from 16.35 to 22.84 FPS on a single H200 GPU, a 39.7% improvement over RF. It also improves five-second VBench quality and semantic scores by 0.29 and 1.25, respectively, and reduces 60-second quality drift from 2.227 to 1.700, demonstrating a favorable balance between generation quality, long-horizon stability, and efficiency.

Limitations and future work. First, our current experimental evaluation covers the 1.3B-parameter Wan2.1 backbone at a resolution of 832×480. Extending the training and evaluation of Ms. Forcing to larger models and higher-resolution videos is a natural direction for future work. Second, the KV-cache update at every rolling step (Alg. 1) is necessary for maintaining the autoregressive context, but its current implementation incurs substantial latency. Consequently, the cache-update overhead partially offsets the acceleration gained by reducing DiT computation with MSP and MSSA. Optimizing this required update could allow the reduction in DiT computation to translate more fully into end-to-end throughput gains.

AI USE STATEMENT

In this work, we used large language models to polish the manuscript for clarity and readability and AI coding tools to assist with running experiments. Our use of AI was limited to these two tasks. We have not used generative AI tools for literature retrieval or discovery, research ideation, drafting sections of the paper, generating synthetic datasets, or proving mathematical claims. The authors reviewed all AI-assisted work. We take full responsibility for the final content of this work, including all text, claims, and artifacts produced with the assistance of AI.

ETHICS STATEMENT

Ms. Forcing aims to improve the computational efficiency of streaming video generation. More accessible video generation can support creative applications, but can also lower the cost of producing deceptive media, impersonation, and non-consensual content. The method builds on pretrained video models and may inherit biases, harmful associations, and privacy or copyright concerns associated with their training data. Our evaluations measure generation quality, temporal consistency, and throughput; they do not establish fairness or safety in deployment. Responsible deployment should therefore include safeguards appropriate to the application, such as content moderation, disclosure of synthetic provenance, and consent protections for depictions of identifiable people, as well as respect for the usage restrictions of underlying models and data.

REPRODUCIBILITY STATEMENT

We describe the multi-scale patchification and attention mechanisms in the methodology section and provide the H-DMD training procedure in Algorithm 1. Section 5.1 identifies the base model and initialization sources, summarizes training settings, and specifies the evaluation prompts, video lengths, metrics, quality-drift calculation, and hardware used for throughput measurement. Appendix A provides the patch and noise schedules, coarse-projection initialization, adapter configuration, positional encoding, warm-up procedure, and DMD optimization settings. Appendix B documents the paired zero-shot probe protocol, including prompt selection, the shared random seed, and the LPIPS calculation. These descriptions provide the implementation and evaluation details needed to reproduce the reported experimental setup.

REFERENCES

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first International Conference on Machine Learning*, 2024.

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

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

Jiayu Chen, Junbei Tang, Wenbiao Zhao, Maoliang Li, Jiayi Luo, Zihao Zheng, Jiawei Yang, Guojie Luo, and Xiang Chen. Pyramid forcing: Head-aware pyramid kv cache policy for high-quality long video generation. *arXiv preprint arXiv:2605.13111*, 2026.

Xiaoyan Cong, Haotian Yang, Angtian Wang, Yizhi Wang, Yiding Yang, Canyu Zhang, and Chongyang Ma. Viva: Vlm-guided instruction-based video editing with reward optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34364–34374, 2026.

Kaixin Ding, Xi Chen, Sihui Ji, Yuan Gao, Liang Hou, Xin Tao, and Hengshuang Zhao. Surf: Signature-retained fast video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9171–9181, 2026.

Hongyang Du, Junjie Ye, Xiaoyan Cong, Runhao Li, Jingcheng Ni, Aman Agarwal, Zeqi Zhou, Zekun Li, Randall Balestriero, and Yue Wang. Videogpa: Distilling geometry priors for 3d-consistent video generation. *arXiv preprint arXiv:2601.23286*, 2026.

Zelin Gao, Qiuyu Wang, Jiapeng Zhu, Jingye Chen, Zichen Liu, Qingyan Bai, Jiahao Wang, Yufeng Yuan, Hanlin Wang, Yichong Lu, et al. Infinite worlds with versatile interactions. *arXiv preprint arXiv:2607.07534*, 2026.

Hang Guo, Zhaoyang Jia, Jiahao Li, Bin Li, Yuanhao Cai, Jiangshan Wang, Yawei Li, and Yan Lu. Efficient autoregressive video diffusion with dummy head. *arXiv preprint arXiv:2601.20499*, 2026.

Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026.

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

Lianghua Huang, Zhi-Fan Wu, Yupeng Shi, Wei Wang, Mengyang Feng, Cheng Yu, Chen Liang, Junjie He, Chen-Wei Xie, Yu Liu, et al. Video = world+ event stream. *arXiv preprint arXiv:2607.15038*, 2026.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 38:167283–167308, 2025.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, 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.

Yang Jin, Zhicheng Sun, Ningyuan Li, Kun Xu, Hao Jiang, Nan Zhuang, Quzhe Huang, Yang Song, Yadong Mu, and Zhouchen Lin. Pyramidal flow matching for efficient video generative modeling. In *International Conference on Learning Representations*, volume 2025, pp. 23378–23402, 2025.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Dahye Kim, Deepti Ghadiyaram, and Raghudeep Gadde. Ddit: Dynamic patch scheduling for efficient diffusion transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11459–11471, 2026.

Jihwan Kim, Junoh Kang, Jinyoung Choi, and Bohyung Han. Fifo-diffusion: Generating infinite videos from text without training. In *Advances in Neural Information Processing Systems*, 2024.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Zekun Li, Rui Zhou, Rahul Sajnani, Xiaoyan Cong, Daniel Ritchie, and Srinath Sridhar. Genhsi: Controllable generation of human-scene interaction videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 138–149, 2026.

Feng Liu, Shiwei Zhang, Xiaofeng Wang, Yujie Wei, Haonan Qiu, Yuzhong Zhao, Yingya Zhang, Qixiang Ye, and Fang Wan. Timestep embedding tells: It's time to cache for video diffusion model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7353–7363, 2025a.

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with taylorseers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15853–15863, 2025b.

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.

Yunhong Lu, Yanhong Zeng, Haobo Li, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jiapeng Zhu, Hengyuan Cao, Zhipeng Zhang, Xing Zhu, et al. Reward forcing: Efficient streaming video generation with rewarded distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34385–34397, 2026.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Ankit Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720*, 2024.

David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. In *International Conference on Machine Learning*, 2024.

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

Joonghyuk Shin, Zhengqi Li, Richard Zhang, Jun-Yan Zhu, Jaesik Park, Eli Shechtman, and Xun Huang. Motionstream: Real-time video generation with interactive motion controls. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=v1DKz5Vxr7.

Mingzhen Sun, Weining Wang, Gen Li, Jiawei Liu, Jiahui Sun, Wanquan Feng, Shanshan Lao, SiYu Zhou, Qian He, and Jing Liu. Ar-diffusion: Asynchronous video generation with auto-regressive diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 7364–7373, 2025.

Zhiyao Sun, Ziqiao Peng, Yifeng Ma, Yi Chen, Zhengguang Zhou, Zixiang Zhou, Guozhen Zhang, Youliang Zhang, Yuan Zhou, Qinglin Lu, et al. Streamavatar: Streaming diffusion models for real-time interactive human avatars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10887–10897, 2026.

Meituan LongCat Team, Xunliang Cai, Qilong Huang, Zhuoliang Kang, Hongyu Li, Shijun Liang, Liya Ma, Siyu Ren, Xiaoming Wei, Rixu Xie, et al. Longcat-video technical report. *arXiv preprint arXiv:2510.22200*, 2025.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

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.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025.

Jung Yi, Wooseok Jang, Paul Hyunbin Cho, Jisu Nam, Heeji Yoon, and Seungryong Kim. Deep forcing: Training-free long video generation with deep sink and participative compression. In *International Conference on Machine Learning*, 2026.

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

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024b.

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

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Xingyu Zheng, Xianglong Liu, Yifu Ding, Weilun Feng, Junqing Lin, Jinyang Guo, and Haotong Qin. Multi-resolution flow matching: Training-free diffusion acceleration via staged sampling. *arXiv preprint arXiv:2607.01642*, 2026.

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

A Additional Implementation Details

Patch and noise schedules. The native patch size in latent coordinates is $(1, 2, 2)$, where the first dimension is temporal. Our five-step schedule uses spatial patch sizes $(p, p, 2p, 2p, 2p)$ from the lowest to the highest source noise level, with $p = 2$. Each chunk contains three latent frames. At a latent spatial resolution of 60×104, the native and coarse token grids per frame are 30×52 and 15×26, respectively. The unshifted timestep schedule is $(1000, 800, 600, 400, 200)$ in denoising order. For $u = t/1000$, we apply the shift $u' = 5u/(1 + 4u)$; the corresponding source timesteps are approximately $(1000, 952.38, 882.35, 769.23, 555.56)$.

Coarse projections and adapters. We initialize the coarse patch embedding by bilinearly interpolating the native spatial kernel and rescaling its Frobenius norm to match the original kernel. The de-embedding weights are spatially interpolated with their mean output-row norm preserved. Rank-32 LoRA adapters are applied to both linear layers of the feed-forward network in every Transformer block, with scaling $\alpha/r = 1$. The first low-rank factor uses Kaiming initialization and the second is zero-initialized. The patch-size embedding and the $(1, 3, 3)$ residual convolution are also zero-initialized; native-tier tokens bypass these coarse-tier adapters.

Spatial positional encoding. A coarse token with spatial stride s is positioned at the centroid $si + (s-1)/2$ on each spatial axis of the native token grid. For a spatial RoPE frequency ω, the implementation multiplies its complex rotation by $s^{-1} \sum_{m=0}^{s-1} \cos(\omega(m-(s-1)/2))$. This averages the positional phases within the represented patch. Temporal RoPE is unchanged.

Adapter warm-up. The warm-up stage freezes the pretrained backbone and updates the coarse projections, patch-size embeddings, residual modules, and LoRA adapters. In a teacher-forced rolling window, it minimizes the mean squared error between the coarse-tier student flow and the frozen native-tier teacher flow after spatial average pooling and nearest-neighbor upsampling. Only coarse-tier chunks contribute to this loss. We use 1000 AdamW updates, a global batch size of 8, $(\beta_1, \beta_2) = (0.9, 0.999)$, and weight decay 0.01. The initial learning rate is 10^{-4} and follows cosine decay toward a floor of 5×10^{-6}.

DMD optimization. The generator starts from the ODE-initialized Wan2.1-T2V-1.3B backbone and the warmed-up adapters; both the backbone and adapters are trainable in this stage. The frozen real-score model is Wan2.1-T2V-14B. The fake-score model is Wan2.1-T2V-1.3B, initialized from the Rolling Forcing critic checkpoint. The text encoder and VAE remain frozen. Both optimizers use AdamW with $(\beta_1, \beta_2) = (0, 0.999)$ and weight decay 0.01; their learning rates are 1.5×10^{-6} for the generator and 4×10^{-7} for the fake score. One training step denotes one fake-score update, and the generator is updated once every five steps. The run uses 64 GPUs with one sample per GPU, BF16 mixed precision, and gradient checkpointing. The training configuration requests a randomly sampled base seed, with a rank-dependent offset. The real-score classifier-free guidance scale is 3.0. Generator EMA begins at step 200 with decay 0.99.

Training duration and short-video generation. We train the DMD stage for 1400 steps, corresponding to 1400 fake-score updates and 280 generator updates. For short-video evaluation, we generate 5-second clips from 946 rewritten official VBench prompts, using five random seeds per prompt. Each clip uses 21 latent frames, decoded to 81 pixel frames at 16 FPS.

B Zero-shot Probe Protocol

The motivation experiment in Fig. 2 uses the pretrained Rolling Forcing checkpoint without finetuning. We use the first three prompts in the example-prompt list and reset the random seed to 0 before each paired rollout. Each rollout has 45 latent frames, decoded to 177 pixel frames at 832×480 and 16 FPS. The two perturbations are evaluated separately against the same dense baseline.

We compute AlexNet-based LPIPS separately for each decoded frame, with pixel values mapped from $[0, 1]$ to $[-1, 1]$, and average each frame's value over the three prompts to obtain the plotted curves. These paired comparisons measure deviation from the dense rollout and serve as a motivation experiment for the mixed-granularity design.