ELIMINATING CONTEXT AND TIMESTEP REDUNDANCY FROM AUTOREGRESSIVE VIDEO DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Few-step autoregressive video diffusion enables streaming and interactive generation, but commonly retains full temporal attention in every layer and a fixed denoising budget for every latent block. Our analysis reveals a consistent layer-wise pattern of temporal-context redundancy: similar subsets of Diffusion Transformer layers assign little attention to historical context across the evaluated training stages and datasets, as well as samples and timesteps within distribution matching distillation (DMD). In contrast, latent frames within a video vary in denoising difficulty and require different computational budgets. To address these two forms of redundancy, we introduce Static Pruning and Adaptive Denoising (SPAD), which reduces computation within each forward pass by removing historical cross-block attention from weak-temporal layers and adapts the number of denoising passes to content difficulty. To avoid online attention ranking, we identify weak-temporal layers from offline profiles and remove redundant historical-context computation using a fixed pruning pattern throughout training and inference. To allocate denoising steps on demand, a lightweight confidence head estimates proximity to the clean endpoint from the current trajectory state. We further introduce probabilistic timestep subsampling during DMD, exposing the generator to variable-span transitions and the states they induce to better accommodate inference-time skipping. Evaluations demonstrate that SPAD accelerates both training and inference while alleviating long-horizon degradation, supporting efficient autoregressive video generation without sacrificing visual quality.

1 INTRODUCTION

Autoregressive (AR) video diffusion has emerged as a promising paradigm for streaming, interactive, and long-form video generation (Chen et al., 2024; Yin et al., 2025; Huang et al., 2025; Zhu et al., 2026; Liu et al., 2026b). Distilling pretrained bidirectional video diffusion models into few-step causal generators reduces computation and latency, but redundancy remains in both temporal-context processing and denoising. In standard pipelines, every diffusion Transformer (DiT) layer retains full causal attention throughout AR training, ODE/consistency distillation (CD), distribution matching distillation (DMD) (Yin et al., 2024b;a), and inference, while few-step generation typically assigns the same denoising schedule to every latent block. Our analysis reveals two distinct forms of redundancy. Temporal-context redundancy follows a consistent layer-wise pattern: similar subsets of DiT layers repeatedly assign little attention to historical context across the evaluated training stages and datasets, as well as samples and DMD timesteps, despite computing full causal attention (Fig. 1(a)). This mismatch between limited use of historical context and the computation devoted to it motivates removing historical attention from these weak-temporal layers. Denoising difficulty, in contrast, varies across latent frames within a video (Le et al., 2026; Liu et al., 2026a), so different amounts of computation are required for different frames to approach their clean endpoints (Fig. 1(b)).

Existing acceleration methods do not fully address these complementary demands. Sparse attention, head-wise policies, and KV cache compression reduce the cost of each model forward pass (Xi et al., 2025; Guo et al., 2026; Ji et al., 2026; Lv et al., 2026; Xu et al., 2026a; Shen et al., 2026). Most are designed primarily for inference or require online importance estimation, leaving underexplored whether historical-context redundancy recurs in similar layers and can be removed consistently throughout distillation. Meanwhile, fixed-step distillation assigns all latent blocks the same denoising budget, measured by the number of function evaluations (NFE). Although adaptive methods show

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.

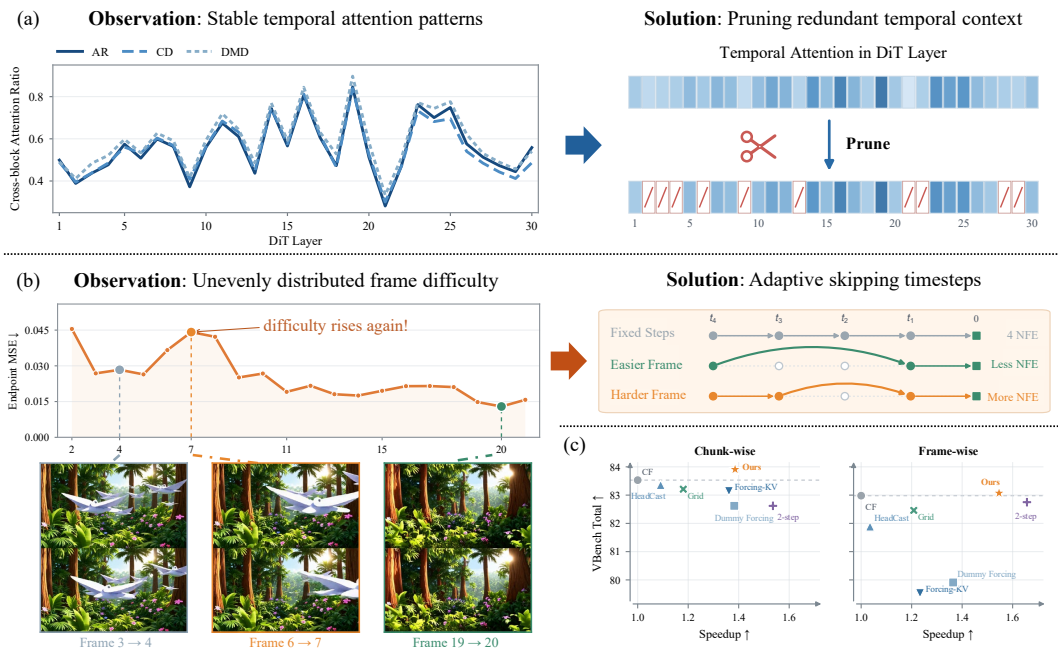


Figure 1: **Context and timestep redundancy** (§1). (a) Similar DiT layers repeatedly assign little attention to historical context across training stages, suggesting a consistent pattern of temporal-context redundancy. (b) Latent frames within the same video exhibit different endpoint errors at a shared denoising timestep, motivating frame-adaptive step allocation. (c) Quality–speed comparisons for 5-second video generation.

that diffusion computation is not equally useful for different samples, frames, or tokens (Tang et al., 2024; Zhang et al., 2025a; Le et al., 2026; Chu & Patel, 2026), inference-time skipping introduces variable-span transitions and intermediate states that a fixed-grid student is not explicitly trained to handle. Efficient AR video diffusion therefore calls for reducing both the cost and the number of model forward passes, with training that supports these changes.

To address the aforementioned context and timestep redundancy, we propose Static Pruning and Adaptive Denoising (SPAD), an efficient AR video diffusion approach that removes historical cross-block attention from weak-temporal DiT layers to reduce computation within each model forward pass and allocates denoising passes according to content difficulty (Fig. 1). SPAD primarily comprises the following three technical contributions. **First**, to remove temporal-context redundancy that recurs in similar layers, we identify weak-temporal DiT layers from offline profiles and restrict their historical attention with a fixed pattern throughout training and inference. The recurring low historical-attention mass in Fig. 1(a) and overlap of weak-temporal layers across datasets, samples, and timesteps in Fig. 2(a,b) motivate targeting the same redundancy throughout distillation, without online attention ranking or input-specific top-k selection. **Second**, to accommodate unequal frame-wise denoising difficulty, we introduce a lightweight confidence head that estimates proximity to the clean endpoint from the current trajectory state and allocates steps on demand. Endpoint error can rise again as a video progresses (Fig. 1(b)), and frames within the same video approach their endpoints at different rates (Fig. 2(c)); neither a uniform schedule nor frame position alone captures this variation. **Third**, to prepare the generator for inference-time skipping, we introduce probabilistic timestep subsampling during DMD. We sample the retained grid size and timesteps to expose the student to variable-span transitions and their on-policy states while preserving the model's original generative capabilities and visual quality. Together, these designs reduce temporal-context computation and unnecessary denoising passes while adapting training to the inference trajectories.

Extensive experiments on 5-second and 30-second video generation demonstrate the effectiveness of SPAD. It improves the quality–efficiency trade-off under both frame-wise and chunk-wise generation, reduces denoising latency by up to 48.9%, and decreases the total training cost of causal distillation by 14.7%. It also substantially alleviates long-horizon degradation, supporting efficient AR video generation over both short and long horizons.

2 Related Work

AR Video Diffusion and Distillation. Diffusion and flow models underpin visual generation (Ho et al., 2020; Song et al., 2021; Karras et al., 2022; Lipman et al., 2023). Video diffusion extends these models to temporal sequences (Ho et al., 2022b; Singer et al., 2023; Ho et al., 2022a), with latent-space formulations improving efficiency (Rombach et al., 2022; Blattmann et al., 2023). DiT backbones (Peebles & Xie, 2023) further support scalable text-to-video generation (Ma et al., 2025a; Yang et al., 2025; Kong et al., 2024; Wan Team et al., 2025). Despite their high generation quality, these models still require multiple sampling iterations, motivating few-step distillation (Salimans & Ho, 2022; Song et al., 2023; Luo et al., 2024; Sauer et al., 2024). DMD (Yin et al., 2024b;a) trains efficient generators through distribution matching. Diffusion Forcing (Chen et al., 2024) combines per-token noise levels with causal sequence modeling, supporting long-video generation (Chen et al., 2025; Sand.ai et al., 2025). CausVid (Yin et al., 2025) uses asymmetric DMD to distill a bidirectional teacher into a causal student. Self Forcing (Huang et al., 2025) trains on self-generated AR rollouts, while Causal Forcing (CF) (Zhu et al., 2026) improves initialization with an AR teacher to reduce architectural mismatch. Causal Forcing++ (Zhao et al., 2026) replaces offline causal ODE regression with online causal CD, learning the AR-conditional endpoint map from local teacher steps without precomputing or storing full trajectories. Recent extensions support real-time long-video generation (Liu et al., 2026b; Yang et al., 2026a). SPAD targets the temporal-context and denoising redundancy that persists across training and inference.

Efficient Attention and Caching for Video Diffusion. Efficient kernels (Dao et al., 2022) and feature caching (Wimbauer et al., 2024; Ma et al., 2024; Zhao et al., 2025; Liu et al., 2025) reduce memory traffic or reuse intermediate computations. Sparse attention exploits spatiotemporal structure and input-stable sparsity patterns (Xi et al., 2025; Zhang et al., 2025b; Li et al., 2025; Luo et al., 2026), while AR-specific methods use head-wise context policies, sparse attention, or KV-cache compression (Guo et al., 2026; Ji et al., 2026; Lv et al., 2026; Xu et al., 2026a; Shen et al., 2026; Samuel et al., 2026). These policies range from one-time head classification to hybrid static pruning and online cache compression, operating at finer granularity than our layer-wise restriction. However, these approaches leave the reuse of a fixed layer-wise pruning pattern across distillation stages underexplored. In contrast, SPAD identifies recurring weak-temporal DiT layers across datasets and distillation stages and reuses one static pruning pattern throughout training and inference.

Adaptive Computation for Diffusion Models. Adaptive diffusion methods optimize sampling schedules and architectures or allocate computation across network depth and samples (Li et al., 2023; Tang et al., 2024; Zhang et al., 2025a). Other approaches use training-free adaptive sampling (Ye et al., 2024), token-level video or patch-level image budgets (Chu & Patel, 2026; Schusterbauer et al., 2026), or arbitrary-interval flow maps (Gu et al., 2026). In AR video generation, step allocation can follow predefined temporal schedules or learned confidence (Liu et al., 2026a; Le et al., 2026). Rather than relying on frame position or teacher-referenced confidence, SPAD supervises confidence using absolute distance to the generator's own full-grid endpoint and couples adaptive denoising with timestep-subsampled DMD training and stage-spanning temporal pruning.

3 Methodology

3.1 Overview and Problem Formulation

We consider a few-step AR video diffusion pipeline that first trains a causal generator, optionally initializes it through ODE regression or CD, and finally performs DMD. Let a video latent contain F frames and be partitioned into $N = F/d$ consecutive blocks, where $d = 1$ for frame-wise generation and $d = 3$ for chunk-wise generation. Let L be the number of DiT layers and $\mathcal{G} = \{\tau_1 > \cdots > \tau_K\}$ be the denoising grid in a uniform pre-warp timestep coordinate. Standard AR generation assigns full causal temporal attention to every layer and all K denoising passes to every block. SPAD relaxes both assumptions: static pruning (§3.2) removes historical cross-block attention from weak-temporal DiT layers using a mask shared across training and inference, while adaptive denoising (§3.3) reduces the number of passes for blocks that have already approached their clean endpoints.

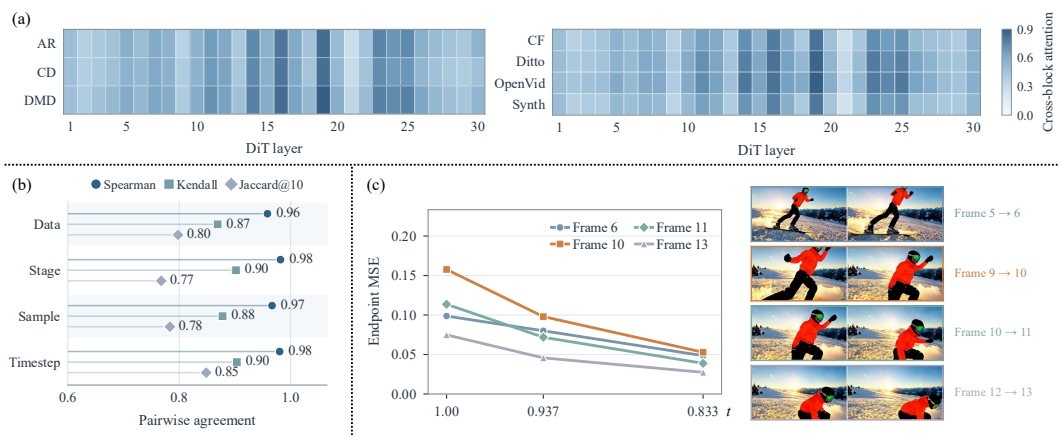


Figure 2: **Recurring temporal-context redundancy and denoising difficulty** (§3.2 and 3.3). (a) Historical attention mass across AR/CD/DMD (left) and CF, Ditto-1M (Bai et al., 2026), OpenVid-1M (Nan et al., 2025), and synthetic data (right); low values identify weak-temporal layers. (b) Spearman and Kendall rank correlations and Jaccard weak-layer overlap measure cross-condition profile agreement. (c) Endpoint MSE for four latent frames versus normalized warped timestep.

3.2 Consistent Temporal-Context Redundancy and Stage-Spanning Pruning

Observation: A recurring layer-wise redundancy pattern. In causal video diffusion models, full temporal attention is computed in every DiT layer, even when a layer assigns little attention to historical blocks. To locate this potential redundancy, we measure the attention mass assigned to history. Let x denote a video latent sample and τ its denoising timestep. At training stage s, $A_{s,\ell}(q,k)$ denotes the self-attention weight from query token q to key token k in layer ℓ, with dependence on x and τ omitted for brevity. For a query block b, we quantify cross-block temporal dependency as

$$r_{s,\ell}(x,\tau,b) = 1 - \frac{\sum_{q\in b}\sum_{k\in b} A_{s,\ell}(q,k)}{\sum_{q\in b}\sum_{k\le b} A_{s,\ell}(q,k)}. \tag{1}$$

Here, $q \in b$ and $k \in b$ index tokens in block b, while $k \le b$ includes keys from the current and preceding blocks. A lower ratio indicates weaker use of historical context. We aggregate over videos, timesteps, and blocks,

$$\bar{r}_{s,\ell} = \mathbb{E}_{x,\tau,b}\left[r_{s,\ell}(x,\tau,b)\right], \tag{2}$$

excluding the first block and auxiliary cache-refresh calls. Fig. 2(a) shows that similar subsets of layers consistently assign little attention to historical blocks across AR/CD/DMD and training datasets. In Fig. 2(b), high Spearman and Kendall rank correlations, together with strong Jaccard overlap of weak-temporal layer sets, indicate consistent profiles across stages, datasets, and DMD samples and timesteps. The three metrics and profiling protocol are detailed in §E. This stable low-attention pattern suggests potential redundancy in historical-context computation, motivating its selective removal through pruning.

Static temporal-attention pruning. To remove historical-context computation from these consistently weak-temporal layers, we average the layer-wise profiles from AR, CD, and DMD with equal weight and select the P lowest-scoring layers, where P is a configurable pruning budget:

$$\bar{r}_\ell = \frac{\bar{r}_{\mathrm{AR},\ell} + \bar{r}_{\mathrm{CD},\ell} + \bar{r}_{\mathrm{DMD},\ell}}{3}, \qquad \mathcal{P} = \mathtt{bottom}_P\left(\{\bar{r}_\ell\}_{\ell=0}^{L-1}\right). \tag{3}$$

Here, $\mathtt{bottom}_P$ returns the indices of the P smallest scores. Both frame-wise and chunk-wise generation reuse this set across AR training, ODE/CD initialization, DMD, and inference; selection is performed once, rather than at each stage or forward pass.

Each selected layer attends only within the current block, preserving full within-block computation while removing historical cross-block attention. The remaining layers keep the original causal mask. During KV-cached inference, selected layers bypass historical KV reads and writes. During teacher forcing, the mask also prevents a noisy block from accessing the clean block at the same position,

avoiding label leakage. By reusing a single offline selection, this design reduces historical-context computation throughout training and inference without online attention ranking, while preserving the model's generative capabilities and visual quality.

3.3 Adaptive Denoising and Skip-Robust DMD

Observation: Unequal denoising difficulty across latent frames. Unlike the consistent layer-wise pattern of temporal-context redundancy, denoising difficulty varies with the generated content (Le et al., 2026; Liu et al., 2026a). Let $x_{t,b}$ be the noisy latent of block b at timestep t, and let $\hat{x}_{0,b}^{(k)}$ be its clean prediction at the k-th grid point. The reference $x_{0,b}^{\text{ref}}$ is a generator prediction at the smallest full-grid timestep. We measure endpoint proximity as

$$e_b^{(k)} = \texttt{mean}\left[\left(\hat{x}_{0,b}^{(k)} - x_{0,b}^{\text{ref}}\right)^2\right], \tag{4}$$

where the mean is taken over all latent frames, channels, and spatial positions in the block.

With adaptive skipping disabled, Fig. 1(b) shows that endpoint error at a fixed denoising timestep rises again at frame 7, rather than decreasing monotonically with frame position. Complementing this observation, Fig. 2(c) shows that latent frames within the same video approach their respective clean endpoints along different trajectories, with some denoising steps yielding only small reductions in endpoint error. Such uneven progress suggests that not every intermediate forward pass contributes equally to denoising. This motivates adapting the next timestep to each frame's current trajectory state, skipping potentially redundant intermediate steps rather than only exiting early once the frame is sufficiently denoised.

Endpoint-proximity confidence. To estimate the remaining denoising demand, we attach a lightweight confidence head to the generator. It consumes the clean prediction $\hat{x}_{0,b}^{(k)}$ and residual $x_{t,b} - \hat{x}_{0,b}^{(k)}$, together with a sinusoidal timestep embedding. The visual input passes through two stride-2 convolutional layers with group normalization, sigmoid linear unit (SiLU) activations, and timestep-conditioned feature-wise linear modulation (FiLM). Spatial mean and standard deviation pooling are concatenated and projected by a two-layer MLP followed by a sigmoid. The head outputs one scalar $c_b^{(k)} \in (0, 1)$ per block. We supervise it with an absolute endpoint-distance target using a fixed normalization scale:

$$c_b^{*(k)} = c_{\max}\left(1 - \texttt{clip}\left(\frac{e_b^{(k)}}{e_{\text{ref}}}, 0, 1\right)\right), \qquad \mathcal{L}_{\text{conf}} = \frac{1}{|\mathcal{Q}|}\sum_{(b,k)\in\mathcal{Q}}\left(c_b^{(k)} - c_b^{*(k)}\right)^2. \tag{5}$$

Here, e_{ref} is a fixed distance scale and $c_{\max} < 1$ bounds the target. The set $\mathcal{Q}$ contains visited pre-exit states from blocks after the first. Detached head inputs and endpoint targets ensure that this loss updates only the confidence head.

Confidence-guided inference. Flow timestep warping makes the model's actual timestep grid highly non-uniform. We therefore perform scheduling in the original uniform coordinate τ. Given the current τ_b and confidence c_b, we compute

$$\tilde{\tau}_b^{\text{next}} = \left(1 - \texttt{clip}(\alpha c_b, 0, 1)\right)\tau_b, \qquad \tau_b^{\text{next}} = \texttt{snap_floor}\left(\tilde{\tau}_b^{\text{next}}, \mathcal{G} \cup \{0\}, \tau_b\right), \tag{6}$$

where α controls the quality–efficiency trade-off. The operation `snap_floor` selects the largest point in $\mathcal{G} \cup \{0\}$ that does not exceed $\tilde{\tau}_b^{\text{next}}$ and is strictly below the current τ_b. A value of zero exits immediately with the current clean prediction. Otherwise, this prediction is re-noised at the corresponding warped timestep for the next forward pass. The first AR block retains the fixed schedule, following the first-frame enhancement principle (Yang et al., 2026b), and subsequent blocks use Eq. (6).

Timestep-subsampled DMD training. A fixed-grid student is not explicitly trained for the states induced by skipped transitions. Each DMD training rollout therefore constructs $\mathcal{G}_{\text{iter}} \subseteq \mathcal{G}$ by drawing its size $n \sim \mathcal{U}\{K_{\min}, \ldots, K\}$ and sampling grid points without replacement, enforcing a full-noise starting point (§G). All workers and blocks after the first share this subset and a uniformly sampled exit position. The generator's DMD gradient is backpropagated through the exit prediction, while confidence supervision uses preceding states.

Table 1: **5-second generation comparisons (§4.2).** Within each generation setting, **bold** and underlining mark the best and second-best values. ↑/↓: higher/lower is better.

	Method	VBench ↑			CF Prompts			Efficiency ↓	
		Total	Quality	Semantic	VR↑	HPS↑	Dyn.	Time (s)	TA
Chunk-wise	Causal Forcing (Zhu et al., 2026)	83.53	85.45	75.82	9.76	9.57	84	11.64	1.00
	Dummy Forcing (Guo et al., 2026)	82.62	84.82	73.80	9.22	8.67	91	8.42	0.73
	HeadCast (Shen et al., 2026)	83.36	85.34	75.43	9.89	9.63	85	10.67	0.92
	Forcing-KV (Ji et al., 2026)	83.14	85.02	75.62	9.78	9.61	90	8.55	**0.42**
	Grid Pruning	83.21	85.45	74.27	10.18	**9.86**	80	9.86	0.67
	Fixed 2-step	82.62	84.87	73.61	9.33	9.50	76	**7.58**	1.00
	SPAD (Ours)	**83.91**	**85.79**	**76.38**	**10.23**	9.53	80	8.40	0.67
Frame-wise	Causal Forcing (Zhu et al., 2026)	82.98	84.93	75.20	9.79	9.31	81	12.06	1.00
	Dummy Forcing (Guo et al., 2026)	79.91	81.40	73.93	8.14	8.93	95	8.84	0.72
	HeadCast (Shen et al., 2026)	81.89	83.76	74.39	7.93	8.85	88	11.65	0.82
	Forcing-KV (Ji et al., 2026)	79.53	81.20	72.84	8.31	8.80	98	9.78	**0.27**
	Grid Pruning	82.46	84.41	74.64	8.87	9.51	49	9.98	0.67
	Fixed 2-step	82.75	84.98	73.81	9.62	9.76	75	**7.28**	1.00
	SPAD (Ours)	**83.07**	**85.03**	**75.23**	**9.73**	**9.96**	77	7.80	0.67

Importantly, the training trajectory is sampled independently of the current confidence prediction. The confidence head observes the states and learns from Eq. (5), but does not determine the next training state in the same rollout. This decoupling avoids direct feedback between confidence predictions and training-trajectory selection while exposing the backbone to skipped trajectories. The generator-update objective is

$$\mathcal{L} = \mathcal{L}_{\text{DMD}} + \lambda_{\text{conf}} \mathbf{1}[u \geq u_{\text{warm}}] \mathcal{L}_{\text{conf}}, \tag{7}$$

where u counts outer training iterations, u_{warm} is the confidence warmup threshold, and λ_{conf} weights the confidence loss.

4 Experiments

4.1 Experimental Setup

Models and training. Experiments are conducted on CF (Zhu et al., 2026) using Wan2.1 text-to-video (T2V) models (Wan Team et al., 2025): Wan2.1-T2V-1.3B as the causal generator and a frozen Wan2.1-T2V-14B as the real-score model. All training data are taken from the open-source CF release. The 5-second setting generates 81 pixel frames at 16 FPS and 480×832 resolution, represented by 21 latent frames. We evaluate both frame-wise ($d = 1$) and chunk-wise ($d = 3$) AR generation. AR training and CD initialization each run for 5,000 steps with a total batch size of 32. DMD converges within 1,000 outer iterations with a total batch size of 16, updating the critic every iteration and the generator every five iterations. Unless otherwise stated, both frame-wise and chunk-wise pipelines use a pruning budget of $P = 10$ out of $L = 30$ DiT layers. The shared mask follows Eq. (3) and remains fixed across AR training, CD initialization, DMD, and inference. The default inference skip strength is $\alpha = 1.0$. More details are provided in §D and G.

For long-video evaluation, we apply SPAD to Rolling Forcing (Liu et al., 2026b) and generate 30-second videos. Additional results on LongLive are reported in §B.

Evaluation protocols. We use three prompt protocols. VBench Total, Quality, and Semantic scores (Huang et al., 2024) are computed on all 946 VBench augmented prompts. Dynamic Degree (Dyn.), VisionReward (VR) (Xu et al., 2026b), and Human Preference Score v3 (HPSv3) (Ma et al., 2025b) are evaluated on the 100 validation prompts released by CF. VR is multiplied by 100 for readability. All long-video metrics are evaluated on 200 MovieGen prompts (Polyak et al., 2024). For long videos, drift is the score difference between the latter and earlier segments. A value closer to zero indicates less degradation over time. Dyn. measures motion magnitude and is therefore reported separately rather than treated as a monotonic perceptual-quality metric. For controlled



Figure 3: 5-second qualitative comparison (§4.2).

Table 2: **30-second generation comparisons (§4.3). Bold** and underlining mark the best and second-best values.

Method	VR↑	VR Drift	HPS↑	HPS Drift	Dyn.	Time (s)↓	TA↓
Rolling Forcing (Liu et al., 2026b)	13.55	-1.38	8.80	-0.73	45	88.26	1.00
Grid Pruning	**13.53**	-1.90	8.94	-1.20	32	70.64	**0.67**
Fixed 2-step	11.40	-3.72	**13.67**	-0.93	28	54.20	1.00
SPAD (Ours) ($\alpha = 1.8$)	13.46	**-0.74**	9.08	**-0.16**	42	**45.08**	**0.67**

comparisons, Grid Pruning restricts an equally sized, uniformly spaced set of DiT layers to within-block attention. Fixed 2-step uses a fixed two-step denoising schedule without temporal pruning. We measure denoising-core latency on one NVIDIA H20 with batch size one, excluding text encoding and variational autoencoder (VAE) decoding. Detailed timing protocols and efficiency-metric definitions are provided in §D.

4.2 Main Results on 5-Second Video Generation

Fig. 1(c) and Tab. 1 show that SPAD improves the quality–efficiency trade-off under both temporal groupings. Relative to the corresponding CF baseline, it reduces latency by 27.8% in chunk-wise generation and 35.3% in frame-wise generation while improving all three VBench aggregate scores.

In the chunk-wise setting, SPAD achieves higher VBench scores than Dummy Forcing and Forcing-KV at comparable latency. In the frame-wise setting, it also attains the highest HPSv3 score among the compared methods. Fixed 2-step sampling is faster in both settings, but yields lower VBench Total and Semantic scores.

Qualitative results. Fig. 3 compares all seven methods under both temporal groupings. In the displayed snowy-forest sequence, SPAD keeps the bird in view across all three frames, whereas it is absent from the final frames of HeadCast and Fixed 2-step. In the beach example, SPAD keeps the corgi's head and chest-mounted camera visible, while Dummy Forcing and Forcing-KV crop parts of the head in later frames. More qualitative results are provided in §H.

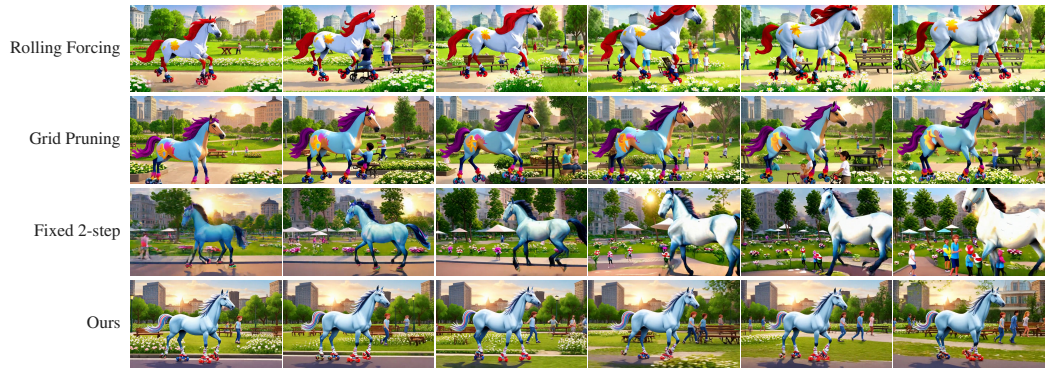


Figure 4: 30-second qualitative comparison (§4.3).

Table 3: **Cumulative component ablations (§4.4).** The final row of each group is the full SPAD model. Within each generation setting, **bold** and underlining mark the best and second-best values.

	Setting	VBench ↑			CF Prompts			Efficiency ↓		
		Total	Quality	Semantic	VR↑	HPS↑	Dyn.	Time (s)	TA	NFE
Chunk-wise	Baseline	83.53	85.45	75.82	9.76	9.57	84	11.64	1.00	4.00
	Grid Pruning	83.21	85.45	74.27	10.18	**9.86**	80	9.86	**0.67**	4.00
	Profiled TP	83.76	**85.82**	75.53	10.20	9.68	81	9.87	**0.67**	4.00
	+ Conf.	83.75	**85.82**	75.48	9.37	9.26	79	9.12	**0.67**	3.51
	+ Subsample (Ours)	**83.91**	85.79	**76.38**	**10.23**	9.53	80	**8.40**	**0.67**	**3.29**
Frame-wise	Baseline	82.98	84.93	75.20	9.79	9.31	81	12.06	1.00	4.00
	Grid Pruning	82.46	84.41	74.64	8.87	9.51	49	9.98	**0.67**	4.00
	Profiled TP	82.97	84.88	**75.33**	9.73	9.63	74	9.91	**0.67**	4.00
	+ Conf.	82.89	**85.09**	74.07	**9.89**	9.95	61	**7.74**	**0.67**	**2.69**
	+ Subsample (Ours)	**83.07**	85.03	75.23	9.73	**9.96**	77	7.80	**0.67**	2.91

4.3 Long-Video Generation

On 30-second generation, SPAD reduces latency from 88.26s to 45.08s (48.9%, 1.96×) and average NFE from 4.00 to 2.15. Dyn. changes from 45 to 42, VR decreases by only 0.09, and HPSv3 improves by 0.28. More importantly, its VR and HPSv3 drifts are closest to zero, indicating less long-horizon degradation under both metrics. Compared with fixed 2-step sampling, SPAD is 16.8% faster, retains more motion as measured by Dyn., and reduces both drift magnitudes, although the fixed baseline obtains a higher absolute HPSv3.

Qualitative results. Fig. 4 visualizes video samples generated by all baseline methods. In this selected roller-skating horse example, SPAD keeps the horse and its colorful skates visible across changing leg poses. The fixed two-step sequence loses recognizable skates and brings the horse partly outside the frame toward the end. More qualitative results are provided in §H.

4.4 Diagnostic Experiments

Component ablations. Tab. 3 presents a cumulative ablation. Starting from profiled temporal pruning (TP), adding confidence-guided inference ("+ Conf.") applies adaptive skipping to a fixed-grid-trained model, while adding probabilistic timestep subsampling ("+ Subsample (Ours)") also adapts its DMD training trajectories, yielding the full SPAD model.

With skipping disabled, profiled TP reduces latency while largely preserving the baseline VBench scores, supporting the removal of historical-context computation from the selected weak-temporal layers. At an identical TA ratio and NFE of 4, profiled TP improves Total/Semantic over Grid Pruning by 0.55/1.26 for chunk-wise generation and by 0.51/0.69 for frame-wise generation, further

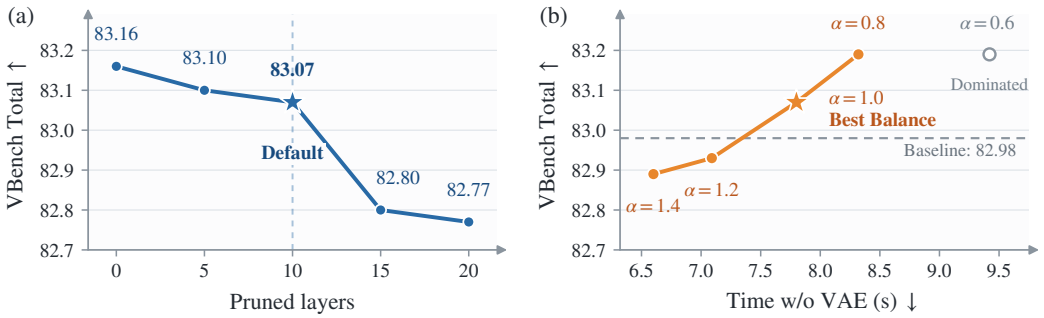


Figure 5: **Pruning and sampling trade-offs** (§4.4). (a) Total versus pruned layers. (b) Total versus latency with ten pruned layers.

supporting the use of these profiles to locate temporal-context redundancy. Confidence-guided inference reduces NFE and latency, but the frame-wise Semantic score drops from 75.33 to 74.07, consistent with a skipped-trajectory mismatch. Probabilistic timestep subsampling then recovers 1.16 Semantic points and improves Total by 0.18, supporting the effectiveness of our training adaptation for inference-time skipping.

Number of pruned layers. Fig. 5a shows the effect of varying the number of pruned layers in frame-wise generation. VBench Total decreases only gradually up to 10 pruned layers, followed by a sharper drop when pruning is increased further. This trend suggests that modest pruning removes largely redundant historical context, whereas more aggressive pruning starts to impair useful temporal information. We therefore select 10 layers to reduce computation with limited quality loss. Additional reward and efficiency results are provided in §C.

Controlling the denoising budget. Fig. 5b illustrates the quality–latency trade-off controlled by α with the pruning mask fixed. Larger values enable more aggressive skipping and faster generation, but can reduce quality. Among the tested settings, $\alpha = 1.0$ is the fastest that preserves the baseline VBench Total score. We therefore select it as the default for the best balance between generation speed and quality.

Training efficiency across distillation stages. Tab. 4 evaluates whether the shared pruning pattern reduces training cost as well as inference latency. Removing historical-context computation from the same weak-temporal layers accelerates every stage, rather than serving only as an inference-time optimization. Per-update time decreases by 15.0%–18.1% for AR training, CD initialization, and the DMD generator, and by 12.2% for the DMD critic. AR and CD together decrease from 60.81 to 51.39 wall-clock hours. With 1,000 DMD outer iterations, the estimated total training time for the frame-wise pipeline decreases from 85.24 to 72.75 hours, a 14.7% reduction.

Table 4: Training time (s/update) at matched stage-wise batch sizes (§4.4).

Stage	Full TA	Pruned	Reduction
AR training	15.96	13.36	16.3%
CD initialization	27.82	23.64	15.0%
DMD generator	25.48	20.88	18.1%
DMD critic	82.88	72.74	12.2%

5 Conclusion and Discussion

In this work, we propose SPAD, an efficient approach to reducing context and timestep redundancy in AR video diffusion. It first identifies weak-temporal layers for static historical-attention pruning and then allocates denoising passes according to endpoint proximity. Probabilistic timestep subsampling further prepares the generator for inference-time skipping. Extensive experiments validate the effectiveness of SPAD in accelerating training and inference while maintaining generation quality and alleviating long-horizon degradation. We believe that SPAD can advance practical interactive video generation by allocating computation according to context and denoising demands.

Limitations. The profiled layer set may depend on the backbone and resolution, and a small calibration set is still required for a new architecture. In addition, overestimated confidence can cause an irreversible aggressive exit. Future work may explore transferable temporal profiles and uncertainty-aware fallback strategies for adaptive denoising.

AI USE STATEMENT

AI tools provided limited support for writing refinement and research execution, specifically language polishing and assistance with code implementation. The work was primarily carried out by the authors, who led the research, made methodological decisions, designed and conducted experiments, and interpreted the results. AI-assisted text and code were reviewed and verified by the authors. This assistive use did not replace human scientific judgment or serve as experimental evidence. The authors retain full responsibility for the code, results, and final manuscript.

ETHICS STATEMENT

We acknowledge the ICLR Code of Ethics and the authors' responsibility to adhere to it. This work studies computational efficiency using existing video-generation models and datasets and does not involve human-subject experiments. More efficient generation may improve accessibility and reduce computational cost, but can also lower the cost of misleading media, non-consensual depictions, and copyright or privacy violations. Biases and limitations of the underlying models and data are not resolved by acceleration. Responsible use should respect applicable licenses, consent and privacy requirements, and use provenance or disclosure mechanisms for generated content. We disclose AI assistance above and clearly distinguish measured results, schematic diagrams, and curated qualitative examples.

REPRODUCIBILITY STATEMENT

We describe the method and experimental setup in §3 and 4.1, with additional implementation and evaluation details in the appendix. Together, these provide the settings needed to reproduce the reported comparisons. Implementation code and configurations are planned for release.

REFERENCES

Qingyan Bai, Qiuyu Wang, Hao Ouyang, Yue Yu, Hanlin Wang, Wen Wang, Ka Leong Cheng, Shuailei Ma, Yanhong Zeng, Zichen Liu, Yinghao Xu, Yujun Shen, and Qifeng Chen. Scaling instruction-based video editing with a high-quality synthetic dataset. In *CVPR*, 2026.

Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, Varun Jampani, and Robin Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. In *NeurIPS*, 2024.

Guibin Chen, Dixuan Lin, Jiangping Yang, Chunze Lin, Junchen Zhu, Mingyuan Fan, Hao Zhang, Sheng Chen, Zheng Chen, Chengcheng Ma, Weiming Xiong, Wei Wang, Nuo Pang, Kang Kang, Zhiheng Xu, Yuzhe Jin, Yupeng Liang, Yubing Song, Peng Zhao, Boyuan Xu, Di Qiu, Debang Li, Zhengcong Fei, Yang Li, and Yahui Zhou. SkyReels-V2: Infinite-length film generative model. *arXiv preprint arXiv:2504.13074*, 2025.

Ernie Chu and Vishal M. Patel. Not all tokens need 40 steps: Heterogeneous step allocation in diffusion transformers for efficient video generation. *arXiv preprint arXiv:2605.06892*, 2026.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *NeurIPS*, 2022.

Yuchao Gu, Guian Fang, Yuxin Jiang, Weijia Mao, Song Han, Han Cai, and Mike Zheng Shou. AnyFlow: Any-step video diffusion model with on-policy flow map distillation. *arXiv preprint arXiv:2605.13724*, 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.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet, and Tim Salimans. Imagen Video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022a.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *NeurIPS*, 2022b.

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

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024.

Yicheng Ji, Zhizhou Zhong, Jun Zhang, Qin Yang, XiTai Jin, Ying Qin, Wenhan Luo, Shuiyang Mao, Wei Liu, and Huan Li. Forcing-KV: Hybrid KV cache compression for efficient autoregressive video diffusion models. *arXiv preprint arXiv:2605.09681*, 2026.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, Kathrina Wu, Qin Lin, Junkun Yuan, Yanxin Long, Aladdin Wang, Andong Wang, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Hongmei Wang, Jacob Song, Jiawang Bai, Jianbing Wu, Jinbao Xue, Joey Wang, Kai Wang, Mengyang Liu, Pengyu Li, Shuai Li, Weiyan Wang, Wenqing Yu, Xinchi Deng, Yang Li, Yi Chen, Yutao Cui, Yuanbo Peng, Zhentao Yu, Zhiyu He, Zhiyong Xu, Zixiang Zhou, Zunnan Xu, Yangyu Tao, Qinglin Lu, Songtao Liu, Dax Zhou, Hongfa Wang, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, and Caesar Zhong. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Thanh-Tung Le, Yunhan Zhao, Menglei Chai, Zhengyang Shen, Zhe Cao, Danhang Tang, Xiaohui Xie, and Deying Kong. DSA: Dynamic step allocation for fast autoregressive video generation. *arXiv preprint arXiv:2606.04432*, 2026.

Lijiang Li, Huixia Li, Xiawu Zheng, Jie Wu, Xuefeng Xiao, Rui Wang, Min Zheng, Xin Pan, Fei Chao, and Rongrong Ji. AutoDiffusion: Training-free optimization of time steps and architectures for automated diffusion model acceleration. In *ICCV*, 2023.

Xingyang Li, Muyang Li, Tianle Cai, Haocheng Xi, Shuo Yang, Yujun Lin, Lvmin Zhang, Songlin Yang, Jinbo Hu, Kelly Peng, Maneesh Agrawala, Ion Stoica, Kurt Keutzer, and Song Han. Radial attention: $O(n \log n)$ sparse attention with energy decay for long video generation. In *NeurIPS*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

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 *CVPR*, 2025.

Jinxiu Liu, Xuanming Liu, Kangfu Mei, Yandong Wen, Ming-Hsuan Yang, and Weiyang Liu. Streaming autoregressive video generation via diagonal distillation. In *ICLR*, 2026a.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling forcing: Autoregressive long video diffusion in real time. In *ICLR*, 2026b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019.

Jiayi Luo, Jiayu Chen, Jiankun Wang, Cong Wang, Hanxin Zhu, Qingyun Sun, Chen Gao, Zhibo Chen, and Jianxin Li. Attention sparsity is input-stable: Training-free sparse attention for video generation via offline sparsity profiling and online QK co-clustering. *arXiv preprint arXiv:2603.18636*, 2026.

Simian Luo, Yiqin Tan, Longbo Huang, Jian Li, and Hang Zhao. Latent consistency models: Synthesizing high-resolution images with few-step inference. In *ICLR*, 2024.

Chengtao Lv, Yumeng Shi, Yushi Huang, Ruihao Gong, Shen Ren, and Wenya Wang. Light forcing: Accelerating autoregressive video diffusion via sparse attention. *arXiv preprint arXiv:2602.04789*, 2026.

Xin Ma, Yaohui Wang, Xinyuan Chen, Gengyun Jia, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *Transactions on Machine Learning Research*, 2025a.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. DeepCache: Accelerating diffusion models for free. In *CVPR*, 2024.

Yuhang Ma, Yunhao Shui, Xiaoshi Wu, Keqiang Sun, and Hongsheng Li. HPSv3: Towards wide-spectrum human preference score. In *ICCV*, 2025b.

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. OpenVid-1M: A large-scale high-quality dataset for text-to-video generation. In *ICLR*, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023.

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. *arXiv preprint arXiv:2410.13720*, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *ICLR*, 2022.

Dvir Samuel, Issar Tzachor, Matan Levy, Michael Green, Gal Chechik, and Rami Ben-Ari. Fast autoregressive video diffusion and world models with temporal cache compression and sparse attention. *arXiv preprint arXiv:2602.01801*, 2026.

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. *arXiv preprint arXiv:2505.13211*, 2025.

Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. Adversarial diffusion distillation. In *ECCV*, 2024.

Johannes Schusterbauer, Ming Gui, Yusong Li, Pingchuan Ma, Felix Krause, and Björn Ommer. Denoising, fast and slow: Difficulty-aware adaptive sampling for image generation. In *CVPR*, 2026.

Jinliang Shen, Lianghao Su, Zheming Li, Kang He, ZiLiang Lai, Yanbing Jiang, and Chengru Song. HeadCast: Casting attention heads for efficient autoregressive video generation. *arXiv preprint arXiv:2607.20125*, 2026.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-A-Video: Text-to-video generation without text-video data. In *ICLR*, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *ICML*, 2023.

Shengkun Tang, Yaqing Wang, Caiwen Ding, Yi Liang, Yao Li, and Dongkuan Xu. AdaDiff: Accelerating diffusion models through step-wise adaptive computation. In *ECCV*, 2024.

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

Felix Wimbauer, Bichen Wu, Edgar Schoenfeld, Xiaoliang Dai, Ji Hou, Zijian He, Artsiom Sanakoyeu, Peizhao Zhang, Sam Tsai, Jonas Kohler, Christian Rupprecht, Daniel Cremers, Peter Vajda, and Jialiang Wang. Cache me if you can: Accelerating diffusion models through block caching. In *CVPR*, 2024.

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

Boxun Xu, Yuming Du, Zichang Liu, Siyu Yang, Ziyang Jiang, Siqi Yan, Rajasi Saha, Albert Pumarola, Wenchen Wang, and Peng Li. Sparse forcing: Native trainable sparse attention for real-time autoregressive diffusion video generation. *arXiv preprint arXiv:2604.21221*, 2026a.

Jiazheng Xu, Yu Huang, Jiale Cheng, Yuanming Yang, Jiajun Xu, Yuan Wang, Wenbo Duan, Shen Yang, Qunlin Jin, Shurun Li, Jiayan Teng, Zhuoyi Yang, Wendi Zheng, Xiao Liu, Dan Zhang, Ming Ding, Xiaohan Zhang, Xiaotao Gu, Shiyu Huang, Minlie Huang, Jie Tang, and Yuxiao Dong. VisionReward: Fine-grained multi-dimensional human preference learning for image and video generation. In *AAAI*, 2026b.

Shuai Yang, Wei Huang, Ruihang Chu, Yicheng Xiao, Yuyang Zhao, Xianbang Wang, Muyang Li, Enze Xie, Yingcong Chen, Yao Lu, Song Han, and Yukang Chen. LongLive: Real-time interactive long video generation. In *ICLR*, 2026a.

Yongqi Yang, Huayang Huang, Xu Peng, Xiaobin Hu, Donghao Luo, Jiangning Zhang, Chengjie Wang, and Yu Wu. Towards one-step causal video generation via adversarial self-distillation. In *ICLR*, 2026b.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, Da Yin, Yuxuan Zhang, Weihan Wang, Yean Cheng, Bin Xu, Xiaotao Gu, Yuxiao Dong, and Jie Tang. CogVideoX: Text-to-video diffusion models with an expert transformer. In *ICLR*, 2025.

Hancheng Ye, Jiakang Yuan, Renqiu Xia, Xiangchao Yan, Tao Chen, Junchi Yan, Botian Shi, and Bo Zhang. Training-free adaptive diffusion with bounded difference approximation strategy. In *NeurIPS*, 2024.

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 *NeurIPS*, 2024a.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *CVPR*, 2024b.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T. Freeman, Frédo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *CVPR*, 2025.

Hui Zhang, Zuxuan Wu, Zhen Xing, Jie Shao, and Yu-Gang Jiang. AdaDiff: Adaptive step selection for fast diffusion models. In *AAAI*, 2025a.

Peiyuan Zhang, Yongqi Chen, Haofeng Huang, Will Lin, Zhengzhong Liu, Ion Stoica, Eric Xing, and Hao Zhang. VSA: Faster video diffusion with trainable sparse attention. In *NeurIPS*, 2025b.

Min Zhao, Hongzhou Zhu, Kaiwen Zheng, Zihan Zhou, Bokai Yan, Xinyuan Li, Xiao Yang, Chongxuan Li, and Jun Zhu. Causal Forcing++: Scalable few-step autoregressive diffusion distillation for real-time interactive video generation. *arXiv preprint arXiv:2605.15141*, 2026.

Xuanlei Zhao, Xiaolong Jin, Kai Wang, and Yang You. Real-time video generation with pyramid attention broadcast. In *ICLR*, 2025.

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 *ICML*, 2026.

Appendix Overview

The appendix begins with preliminary background on causal video distillation (§A), followed by LongLive and pruning-strength results (§B and C). Implementation details, observation protocols, mask construction, and adaptive scheduling are provided in §D–G. Additional visualizations are presented in §H.

A Preliminary: Causal Distillation for Video Generation

Distilling a pretrained bidirectional video diffusion model into a few-step causal generator involves two changes: replacing access to future frames with causal conditioning, and reducing the number of denoising passes. We summarize the three-stage pipeline of Causal Forcing and Causal Forcing++ (Zhu et al., 2026; Zhao et al., 2026), which separates AR adaptation, few-step initialization, and distribution matching.

Stage 1: Autoregressive adaptation. Starting from pretrained bidirectional weights, the model is fine-tuned with a causal attention mask to denoise each latent block using only its own noisy state and preceding clean blocks. For N blocks $x_{0,1:N}$ and text condition y, the resulting model factorizes the video distribution as

$$p_\theta(x_{0,1:N} \mid y) = \prod_{b=1}^{N} p_\theta(x_{0,b} \mid x_{0,<b}, y). \tag{8}$$

Here, each block contains one or several latent frames. Teacher-forced diffusion training uses clean blocks from the training data as history and the standard denoising or flow-matching objective for the current block. This stage produces a many-step AR diffusion teacher. A causal mask alone does not establish few-step sampling capability.

Stage 2: Few-step initialization. The AR teacher next initializes a few-step causal student. In causal ordinary differential equation (ODE) distillation, the teacher generates probability-flow ODE

trajectories conditioned on clean history. The student regresses noisy intermediate states to the corresponding teacher endpoints under the same history. Causal Forcing (Zhu et al., 2026) uses an AR teacher for this stage because trajectories from a bidirectional teacher can depend on future frames unavailable to the student, creating a mismatch in the regression target.

Causal consistency distillation (CD) is an alternative to this offline trajectory regression (Song et al., 2023; Zhao et al., 2026). An online AR-teacher ODE step connects two nearby noisy states under the same clean history. The student's endpoint prediction at one state is matched to a stop-gradient, moving-average student's prediction at the other. Causal Forcing++ uses this local consistency supervision to learn the AR-conditional endpoint map without precomputing and storing full teacher trajectories. Thus, ODE regression and causal CD are alternative initialization routes, not consecutive stages. Our experiments use CD initialization.

Stage 3: Distribution matching on causal rollouts. The initialized student is refined with DMD (Yin et al., 2024b;a). Following Self Forcing (Huang et al., 2025), it generates videos block by block using its own previous outputs as context, exposing training to the imperfect histories encountered during inference. DMD perturbs the generated videos and compares a frozen real-score model's estimate of the noisy data score with a trainable fake-score model's estimate of the noisy student score. Their difference supplies the generator's distribution-matching gradient, while the fake-score model is trained on generated samples. Unlike ODE/CD initialization, this stage matches distributions rather than paired teacher trajectories. The score models can remain bidirectional even though the student is causal, as in asymmetric DMD (Yin et al., 2025; Zhu et al., 2026). The AR teacher used for initialization is therefore distinct from the real-score teacher used by DMD.

Within this pipeline, SPAD applies its static temporal-attention mask across all three stages, while confidence supervision and probabilistic timestep subsampling are introduced during DMD.

B LongLive-Backbone Comparison

Table 5: **30-second LongLive comparison (§B).** Evaluated on 200 MovieGen prompts. **Bold** and underlining mark the best and second-best values, excluding the gray reference baseline.

Method	VR↑	HPS↑	Dyn.	Time (s)↓	TA↓	NFE↓
LongLive (Yang et al., 2026a)	13.69	7.71	86	71.98	1.00	4.00
Grid Pruning	**14.12**	**8.90**	52	62.14	**0.67**	4.00
Fixed 2-step	13.42	7.88	75	**43.36**	1.00	**2.00**
SPAD (Ours)	13.90	8.24	89	45.18	**0.67**	2.64

Tab. 5 shows that SPAD reduces latency by 37.2% over LongLive while improving both VR and HPSv3. Compared with Grid Pruning, it trades somewhat lower reward scores for faster generation and substantially higher Dyn. Fixed 2-step sampling is slightly faster, but yields lower reward scores and Dyn., highlighting the quality–efficiency trade-off offered by SPAD.

Qualitative comparison. Figs. 6 and 7 illustrates four 30-second examples. In the panda and ancient-tree cases, SPAD keeps the guitar-playing character and the tree's trunk and branching structure recognizable across the displayed frames. The dandelion retains its seed-head structure, while the walking scene keeps the woman visible against the icy landscape as the framing changes.

C Additional Pruning-Strength Results

Tab. 6 shows how the pruning budget affects reward scores, motion magnitude, and efficiency. The 10-layer setting achieves the highest HPSv3 while keeping VR and Dyn. close to the baseline. More aggressive pruning substantially reduces Dyn.; pruning 20 rather than 15 layers increases NFE without further reducing latency. Since nonzero settings also enable adaptive skipping, these timings reflect their combined effect rather than pruning alone.



Prompt: A vibrant and lively illustration in a cartoon style of a panda playing the guitar. The panda has black and white fur, with round eyes and a friendly expression. It sits comfortably on a small stool, strumming the guitar with one paw while the other rests on its knee. The guitar is a small acoustic model, and the strings are plucked with precision. The background features a cozy room with a few plants and colorful decorations, adding a warm and inviting atmosphere. The camera angle is slightly from above, capturing the panda's joyful and focused performance.



Prompt: A low-angle perspective of an ancient tree with gnarled, weathered roots, standing tall and imposing. The tree's bark is rough and deeply textured, with patches of moss and lichen clinging to its surface. Its branches stretch out wide, adorned with sparse, emerald-green leaves. The ground around the tree is covered in a carpet of fallen leaves and twigs, adding to the ancient and serene atmosphere. The sky above is a mix of deep blues and grays, with wisps of clouds drifting by. A close-up shot from a low angle, emphasizing the tree's majestic presence.

Figure 6: Qualitative comparisons on the LongLive backbone (§B).

Table 6: **Pruning-strength comparison (§C).** Nonzero pruning includes adaptive skipping, so runtime does not isolate pruning alone. **Bold** and underlining mark the best and second-best values.

Pruned layers	VR↑	HPS↑	Dyn.	Time (s)↓	TA↓	NFE↓
0 (fixed baseline)	9.79	9.31	81	12.06	1.00	4.00
5	**9.86**	9.73	80	9.40	0.83	3.30
10 (default)	9.73	**9.96**	77	7.80	0.67	2.91
15	<u>9.81</u>	<u>9.81</u>	41	**5.98**	<u>0.50</u>	**2.40**
20	9.11	9.71	51	<u>6.01</u>	**0.33**	<u>2.71</u>

D Additional Implementation Details

Optimization. We use flow timestep shift 5.0, guidance scale 3.0, generator learning rate 2×10^{-6}, critic learning rate 4×10^{-7}, AdamW (Loshchilov & Hutter, 2019) with $\beta_1 = 0$, $\beta_2 = 0.999$, and exponential moving average (EMA) decay 0.99. The four-point pre-warp grid is $\{1000, 750, 500, 250\}$. During DMD training, the first block retains a fixed grid but is randomly truncated for the update. The selected zero-based layer indices are $\{1, 2, 3, 5, 8, 12, 20, 21, 27, 28\}$.



Prompt: A hyper-realistic macro photograph capturing the intricate details of a dandelion, zooming in at an incredible speed to reveal a dream-like, abstract world. The petals are softly blurred, creating a mesmerizing effect that blends reality with fantasy. Each fiber and grain of pollen is vividly detailed, giving the image a surreal texture. The background fades into a gradient of soft pastel colors, enhancing the ethereal quality of the scene. The dandelion appears almost otherworldly, with its delicate structure and vibrant colors standing out against the blurred, abstract surroundings. A close-up shot with a dynamic zoom-in motion.



Prompt: A dramatic winter storm in Antarctica captures a woman taking a pleasant stroll. She wears blue jeans and a white t-shirt, her clothes billowing slightly in the strong winds. Her expression is serene and determined as she walks confidently across the icy landscape. The storm clouds are dark and ominous, with snow swirling around her. In the background, towering ice formations and jagged glaciers add to the harsh yet beautiful setting. The photo has a documentary-style texture, emphasizing the raw power of nature. A dynamic mid-shot from a low-angle perspective, capturing the woman's natural movement and the vast, icy environment.

Figure 7: Further LongLive comparisons (§B).

Confidence optimization. We use $c_{\max} = 0.95$ and $e_{\text{ref}} = 0.06$. The output bias is initialized to -1.1, yielding an initial confidence near 0.25 and a conservative schedule. For frame-wise DMD, the confidence warmup is 50 iterations and its learning-rate multiplier is 50; for chunk-wise DMD, they are 20 and 100, respectively. The confidence loss weight is $\lambda_{\text{conf}} = 5.0$. The head has a separate optimizer and receives supervision from generator-produced rollouts used for either generator or critic updates, with the same loss weight and warmup threshold. Timestep subsampling remains active during confidence warmup. The live confidence head is used with the EMA generator because applying the slow backbone EMA to a newly initialized head lags its learned schedule.

Efficiency measurement. For every method, we generate 100 videos, discard the first video as warmup, and average the remaining latency measurements under the setup in §4.1. The temporal-attention (TA) ratio, $(L - P)/L$, is the fraction of layers retaining full temporal attention. We report NFE as the average number of denoising forward passes per generated block, which can be non-integer under adaptive scheduling.

E Observation Measurement Protocol

Temporal-attention profiles. The profiles in Fig. 1(a) and Fig. 2(a,b) use frame-wise CF models. We analyze AR, CD, and DMD runs, and separate AR runs trained on CF, Ditto, OpenVid, and Synth data. Each run contains 100 sample profiles. AR profiling follows its 50-step sampler, whereas CD and DMD use 4 denoising points. We exclude the first block, which has no historical context, and auxiliary clean-cache refresh calls. For each layer and denoising call, we uniformly sample up to 64 query tokens, compute softmax attention over the available causal keys, and average the historical attention mass over heads and queries. Normal conditional/unconditional calls are averaged where applicable, followed by averaging over blocks, timesteps, and samples.

Profile stability and recurring redundancy. For two layer-wise temporal-attention profiles $u, v \in \mathbb{R}^L$, Spearman correlation is the Pearson correlation between their layer-score ranks. It measures whether relatively weak- and strong-temporal layers retain similar rankings, even when absolute attention magnitudes differ. Kendall correlation instead compares pairs of layers: a pair is concordant if both profiles order its scores in the same way and discordant if the ordering is reversed. It summarizes the balance between concordant and discordant pairs. Both coefficients range from -1 to 1, with 1 indicating identical rankings and -1 reversed rankings, when there are no ties.

Jaccard agreement focuses on the selected weak-temporal layers rather than the full ranking. Let $B_{10}(u)$ and $B_{10}(v)$ denote the ten lowest-ratio layer sets, matching the default pruning budget. Their overlap is $|B_{10}(u) \cap B_{10}(v)|/|B_{10}(u) \cup B_{10}(v)|$, ranging from 0 for disjoint sets to 1 for identical sets, regardless of ordering within each set. Each entry in Fig. 2(b) averages all unordered profile pairs across the corresponding datasets, stages, DMD samples, or DMD timesteps. Together, high rank correlations and substantial Jaccard overlap indicate both similar overall layer orderings and recurring low-attention layer locations.

Frame-wise endpoint error. For the difficulty analysis, we disable adaptive skipping and record all four clean predictions in each frame's rollout. The reference is the last prediction at warped timestep $t = 625$, not ground truth or a separate many-step teacher output. We compute latent-space MSE without dividing by the initial error or the confidence scale e_{ref}. Fig. 1(b) shows frames 2–21 from one video at the second grid point, highlighting frames 4, 7, and 20. Fig. 2(c) shows frames 6, 10, 11, and 13 from another video. Its horizontal coordinates use the logged integer warped timesteps divided by 1,000: 1.000, 0.937, and 0.833. The reference at 0.625 is omitted because its error is identically zero. This coordinate differs from the uniform pre-warp coordinate used by our scheduler.

The paired images are representative pixel frames from the final decoded video at adjacent latent-frame positions. They illustrate local content changes. Endpoint MSE always compares a frame's own clean prediction with its own reference, not with the preceding frame.

F Temporal-Attention Mask Construction

Following §3.2, a restricted layer allows a query block b to access only tokens within that same block. During teacher forcing, the input is organized as clean blocks followed by noisy blocks. In a restricted layer, each noisy block attends only to itself, excluding all clean blocks, including the clean block at the same position, which prevents target leakage. Unrestricted layers retain the original causal teacher-forcing mask. In KV-cached inference, a restricted layer bypasses historical KV-cache writes and reads. For rolling windows containing several blocks, the same restriction is implemented as block-diagonal self-attention.

G Confidence Estimation and Adaptive Scheduling

The confidence head first averages the concatenated $[\hat{x}_0, x_t - \hat{x}_0]$ representation over the latent frames inside each AR block. Two 3×3 stride-2 convolutions use channel widths 64 and 96. Each convolution is followed by group normalization, SiLU, and FiLM generated from a 128-dimensional sinusoidal timestep embedding. The spatial mean and standard deviation are concatenated and projected by an MLP of hidden width 64. The final sigmoid yields one confidence scalar per block. Thus, the method makes a frame-wise decision for $d = 1$ and shares a decision across each three-frame chunk when $d = 3$.

For each block with supervised pre-exit states, training constructs an endpoint reference from its sampled exit prediction. If the exit is already at the smallest full-grid timestep, its detached prediction is reused. Otherwise, we re-noise that detached prediction to the smallest full-grid timestep and run one auxiliary generator forward without gradients. This reference is used only for confidence supervision, not as the block output for DMD or subsequent AR conditioning. Unlike the profiling reference in §E, it does not require a complete full-grid rollout. The normalization scale e_{ref} is fixed across subsets, but reference values and errors still depend on the sampled trajectory.

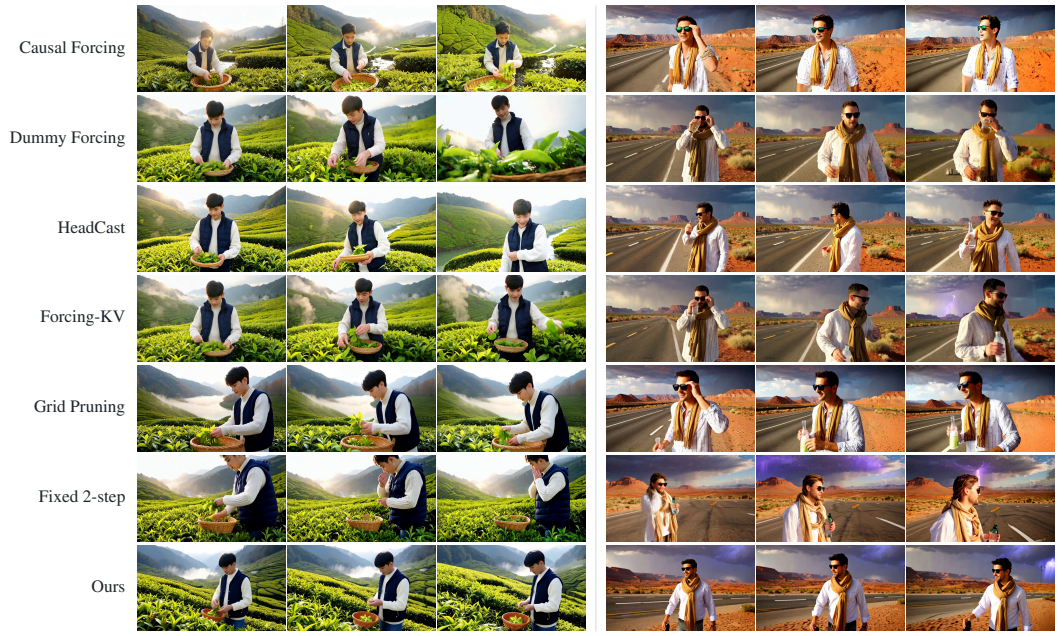
At inference, scheduling is performed in the uniform pre-warp coordinate. The continuous target in Eq. (6) is snapped to the largest grid point that is strictly below the current timestep and does not exceed the target, with zero always available as an exit. This guarantees termination in at most K denoising passes and permits a one-step exit from any grid point. The same per-block rule is used for CF, Rolling Forcing, and LongLive experiments.

For each training rollout, we independently sample a grid subset following §3.3: its size is drawn uniformly between $K_{\min}$ and K, then grid points are sampled uniformly without replacement. If the maximum-noise endpoint is absent, it replaces the smallest sampled timestep, preserving the sampled size and a full-noise starting point. Our main configuration uses $K_{\min} = 3$ and $K = 4$. The subset and a uniformly sampled exit position are shared across workers and blocks after the first. The sampled trajectory, rather than the confidence head, determines the next training state.

H More Visualizations

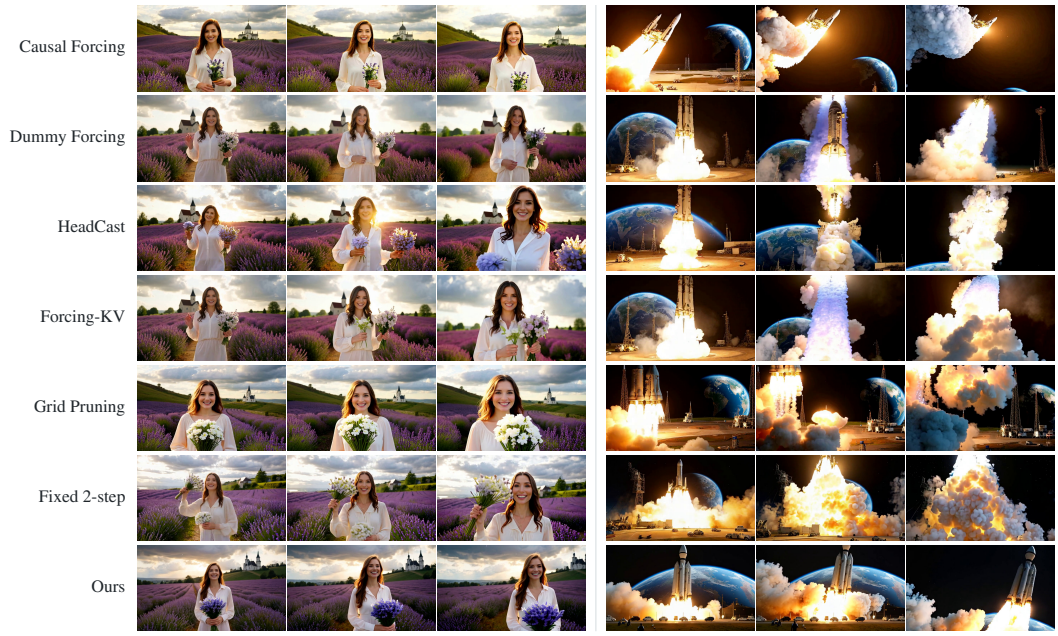
Figs. 8 and 9 show eight 5-second examples under chunk-wise and frame-wise generation. In the chunk-wise cases, SPAD keeps the tea picker and basket, desert traveler, woman and bouquet, and launching shuttle recognizable across the displayed frames. The frame-wise cases cover seaside exercise, skiing, coastal cliffs, and a waterfall, illustrating changes in human pose and water appearance while retaining recognizable subjects and scene structure.

Additional Long-Video Examples. Figs. 10 and 11 complement Fig. 4 with four 30-second MovieGen examples. SPAD keeps the armored hedgehog rider recognizable and the dog and tennis ball in view across the displayed frames. The floating otter remains framed by riverbank vegetation, while the honey-pouring example preserves the visible stream and glossy pancake surface.



Prompt: A young man picking tender tea leaves into a woven basket in a misty mountain tea valley in early spring.

Prompt: A traveler in a white shirt and tan scarf drinking water beside a desert road.



Prompt: A woman holding a fresh bouquet near her face while walking through a sunlit lavender field after rain.

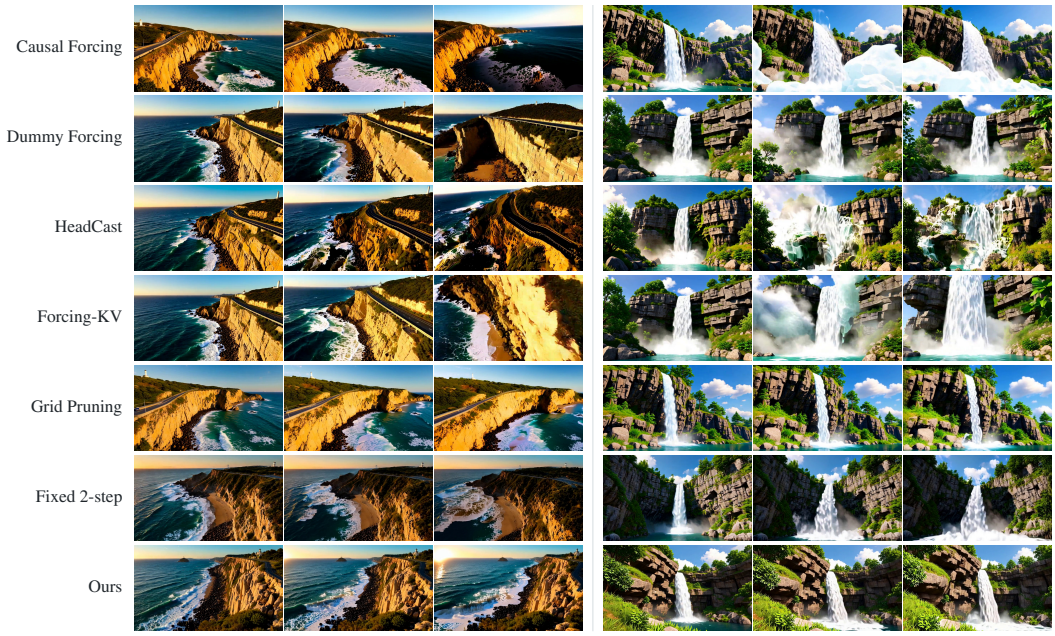
Prompt: A space shuttle lifting off amid flames and smoke, with the blue Earth behind it.

Figure 8: **Additional chunk-wise examples (§H).** Prompts are simplified.



Prompt: A man warming up and stretching on a seaside promenade, with glittering waves and spring flowers.

Prompt: A skier in a red jacket crossing powder snow beneath mountains at dawn.



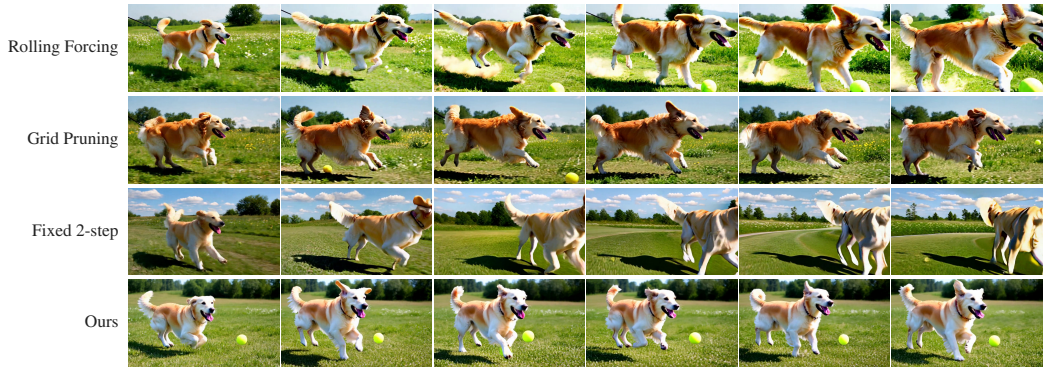
Prompt: A drone view of waves crashing against the cliffs of Big Sur, with green shrubs and golden sunset light.

Prompt: A waterfall flowing down rocky cliffs into a pool amid mist and greenery.

Figure 9: **Additional frame-wise examples (§H).** Prompts are simplified.

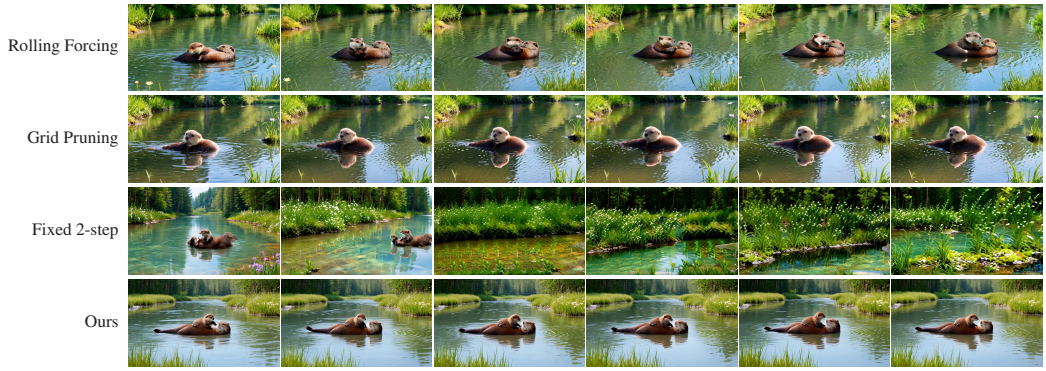


Prompt: A whimsical medieval illustration in a cartoon style, depicting a hedgehog donning intricate knight's armor, complete with a shining helmet and a breastplate adorned with small spikes. The hedgehog rides a tiny toy horse, galloping towards a grand medieval castle with turrets and drawbridges. The castle walls are made of rough stone, with vines and wildflowers growing around the base. The sky is a clear blue with fluffy clouds, and the sun casts a warm, golden light. The scene is alive with movement, capturing the hedgehog's determined charge. A dynamic medium shot from a slightly elevated angle, emphasizing the hedgehog's heroic pose and the toy horse's gallop.



Prompt: A dynamic photograph in the style of action sports imagery, capturing a golden retriever dog sprinting full speed ahead after being released from its leash. The dog's fur glistens in the sunlight, and it runs with intense focus, tail wagging wildly, mouth slightly open, tongue hanging out. Its front legs are extended, and the hind legs powerfully pump, propelling it towards a bright yellow tennis ball rolling away in the distance. The background shows a grassy field with patches of wildflowers, and a few trees in the horizon. The photo has a crisp, high-resolution quality, emphasizing the dog's energetic movement. A mid-shot from a low-angle perspective, capturing the dog's entire body in motion.

Figure 10: Additional Rolling Forcing comparisons (§H).



Prompt: A serene watercolor painting depicting a mother otter floating gracefully on her back in a tranquil river. The otter cradles her playful pup on her stomach, gently keeping it warm and safe in the gentle current. The pup's small paws dangle in the water, while the mother's fur glistens in the soft sunlight. The background features a lush forest with tall trees reflected in the river, and a few wildflowers dotting the banks. The water has a soft, ethereal quality, emphasizing the peacefulness of the scene. A medium shot capturing the tender interaction between the mother and her pup from a slightly overhead angle.



Prompt: A close-up of honey being drizzled onto pancakes, the thick golden liquid flowing slowly and smoothly down the surface. The pancakes are golden brown and fluffy, with a slight steam rising from their edges. The honey forms intricate patterns as it drips, creating a glossy sheen. The background is a warm kitchen setting with a rustic wooden table and a few utensils nearby, adding a cozy and inviting atmosphere. The photo has a soft and natural lighting effect, emphasizing the rich textures and colors. A close-up shot from a slightly tilted angle.

Figure 11: Further Rolling Forcing comparisons (§H).