RETHINKING STREAMING VIDEO DIFFUSION: CONTEXT, EXECUTION, AND TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Understanding the design space of streaming video diffusion is essential to exploring its potential for generation quality and computational efficiency. We develop a unified analytical framework that relates model and sampler choices, historical conditioning, execution scheduling, and training strategies. The framework accommodates a broad family of causal context-selection policies and makes their computational dependencies and training–inference alignment explicit. Within this design space, we study three representative policies: clean, same-level, and progressive history. On the VBench evaluation set, the three policies use rank-64 fake-score LoRA, with same-level and progressive history achieving aggregate scores of 85.24 and 85.60, respectively, compared with 85.16 for clean history. Long-video comparisons further show improved subject consistency and more coherent motion with progressive history. By allowing multiple denoising nodes to be processed together, progressive-history pipelining achieves 1.57–$2.83\times$ steady-state DiT speedups under our evaluated conditions. We additionally find that LoRA adaptation of the DMD fake-score network improves generation quality under clean and progressive history using only 2.15% as many trainable fake-score parameters as full-parameter adaptation. Together, these findings show that fully denoised history is not a prerequisite for high-quality streaming generation and motivate the joint design of historical conditioning, execution, and training.

1 INTRODUCTION

Streaming video generation requires a model to produce temporally coherent content while continuously making new frames available. This requirement differs from generating an entire clip before displaying any output: both the time to the first output and the interval between subsequent outputs matter. Autoregressive video diffusion models provide a natural basis for this setting, combining incremental generation over video chunks with iterative denoising within each chunk (Ho et al., 2022; Henschel et al., 2025). Their execution is therefore governed by dependencies along two axes: video time and denoising progress (Ruhe et al., 2024). Understanding how these dependencies interact is important for improving streaming efficiency without compromising generation quality.

A common design conditions each new chunk on fully denoised historical chunks. Self Forcing (Huang et al., 2025), for example, incorporates autoregressive rollout into training so that the model learns from its own generated history. This addresses a training–inference mismatch in the source of conditioning information. However, a clean-history rollout also makes the next chunk wait for the denoising process of its predecessor to finish. StreamDiffusionV2 (Feng et al., 2026) demonstrates how stream batching and pipelined execution can overlap work across denoising steps and network layers. Motivated by this execution perspective, we examine the denoising state of historical context as a modeling choice that must be considered together with the execution schedule.

Our central question is: *which denoising state should a video chunk use for its historical context, and how does this choice shape generation quality and execution efficiency?* Temporal causality restricts access to future video content, while leaving room to select among the denoising states of historical chunks (Chen et al., 2024). These choices determine both the conditioning information supplied to the model and the dependencies governing when different chunks can advance. Historical context thus connects the modeling and execution aspects of streaming video generation.

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.

We develop a unified analytical framework for streaming video generation that relates model and sampler choices, historical-context construction, execution scheduling, and training strategies. The framework accommodates a broad family of causal context-selection policies and makes explicit their relationships with attention visibility, training-state sources, and supervision objectives. Within this framework, we study three representative context policies: clean, same-level, and progressive history. We examine whether partially denoised history can provide effective conditioning for streaming generation, and how different context choices shape generation quality and inference efficiency. We also discuss empirical observations on parameter-efficient fake-score adaptation and generation quality using LoRA (Hu et al., 2022) during Distribution Matching Distillation (DMD) (Yin et al., 2024b).

With rank-64 fake-score LoRA, same-level and progressive history achieve comparable or higher aggregate scores than clean history on the VBench evaluation set (Huang et al., 2024), with qualitative examples, including long videos, illustrating more coherent subject motion and better subject consistency. Progressive-history pipelining achieves 1.57–$2.83\times$ steady-state DiT speedups under our evaluation conditions. We also find that LoRA fake-score adaptation enables faster distillation and better generation quality using only 2.15% as many trainable parameters as full-parameter adaptation.

Our contributions are twofold:

- **A unified framework for streaming video diffusion.** We characterize a broad family of causal context policies and relate historical-context construction to model and sampler choices, execution scheduling, and training strategies, clarifying their implications for parallelism and training–inference consistency.
- **A framework-guided study of quality and efficiency.** We study the generation quality, long-video behavior, and steady-state inference efficiency of representative context policies, interpreting their behavior through the framework, and discuss parameter-efficient fake-score adaptation during distillation.

2 Related Work

Video Diffusion and Autoregressive Generation. Streaming video generation has advanced along two complementary directions: diffusion-based synthesis and autoregressive sequence modeling. Early video diffusion extended image denoising architectures to spatiotemporal data (Ho et al., 2022). Subsequent progress in latent-space modeling, large-scale training, and spatiotemporal compression enabled increasingly scalable, high-resolution text-conditioned generation (Blattmann et al., 2023b;a; Yang et al., 2025). In parallel, autoregressive approaches represented videos as discrete token sequences, progressing from modeling quantized video latents to multimodal, text-conditioned generation (Yan et al., 2021; Kondratyuk et al., 2024). For streaming synthesis, combining temporal autoregression with diffusion within each video chunk brings incremental generation together with iterative refinement. Within this formulation, causal architectures and few-step distillation reduce generation latency (Yin et al., 2025), while training on model-generated histories helps alleviate the mismatch between training and autoregressive inference (Huang et al., 2025). Building on these developments, our work examines how historical context, execution scheduling, and training interact in streaming video generation.

Noisy Context and Pipelined Execution. The organization of noise levels across video frames constitutes another dimension of streaming video generation. Diffusion Forcing (Chen et al., 2024) trains causal sequence models with independent per-token noise levels, enabling flexible sampling schedules and conditioning on noisy history. History-Guided Video Diffusion (Song et al., 2025) extends this noise-based conditioning approach to a non-causal transformer and constructs guidance by combining scores conditioned on different history subsets and noise levels. FIFO-Diffusion (Kim et al., 2024) performs training-free diagonal denoising over a latent queue, using partitioning and lookahead to manage noise-level mismatch; its partitioned design also supports multi-GPU inference. SkyReels-V2 (Chen et al., 2025) adopts non-decreasing noise schedules for long-video generation and supports synchronous and asynchronous sampling. At the systems level, StreamDiffusionV2 (Feng et al., 2026) combines stream batching, rolling KV caching, and pipeline orchestration across denoising steps and network layers for interactive streaming. These approaches couple noise-

state choices with different conditioning mechanisms, attention structures, and execution schemes. Our framework separates these design elements and uses representative context policies to study their effects on generation quality and inference efficiency.

Distillation and Parameter-Efficient Adaptation. Diffusion distillation reduces sampling cost by learning generators that require fewer denoising steps. Progressive distillation (Salimans & Ho, 2022) repeatedly halves the sampling steps of a deterministic teacher, while consistency models (Song et al., 2023) learn mappings that agree along probability-flow trajectories, enabling one- and few-step generation. Distribution Matching Distillation (DMD) (Yin et al., 2024b) aligns student and teacher output distributions using teacher scores and a learned fake-score model. DMD2 (Yin et al., 2024a) links training instability to inaccurate tracking of the evolving student distribution by the fake-score model and stabilizes optimization through a two-time-scale update rule. Parameter-efficient adaptation offers a complementary direction: LoRA (Hu et al., 2022) parameterizes weight updates with low-rank matrices, and LCM-LoRA (Luo et al., 2023) applies this parameterization to latent consistency distillation of the generator. Our training-side study instead applies low-rank adaptation to the auxiliary fake-score network in DMD, examining how its parameterization affects optimization cost and student generation quality alongside context-policy adaptation.

3 A Unified Framework for Streaming Video Generation

To place historical denoising-state selection in context, we first formulate the streaming video generation process. Our formulation describes how video is partitioned into generation units, how these units are denoised and conditioned, how computation is organized, and how the model is trained.

3.1 Generation Process and Design Elements

Consider a sequence of B latent video chunks conditioned on c. The chunk size determines the temporal granularity of generation. Each chunk follows a noise schedule $0 = t_0 < t_1 < \cdots < t_K$, where x_b^j denotes the state of chunk b at noise level t_j. During generation, x_b^K is the initial noisy state and x_b^0 is the clean endpoint. One denoising update consists of a network prediction followed by a sampling step:

$$y_{b,j} = F_\theta(x_b^j, t_j; \mathcal{C}_{b,j}, M_{b,j}, c), \qquad x_b^{j-1} = S_j(x_b^j, y_{b,j}; \xi_{b,j}). \tag{1}$$

Here F_θ is the denoising network, whose output may parameterize noise, score, clean-sample, or velocity prediction; S_j is the corresponding sampling update, with optional randomness $\xi_{b,j}$. The noise schedule, number of updates, and sampler govern progression within each chunk.

The context $\mathcal{C}_{b,j}$ and attention mask $M_{b,j}$ govern interaction between chunks. Constructing them involves selecting historical states and maintaining their cached representations. The execution pattern determines which chunks can advance together. Training further specifies the sources of input states, the supervision signal, and the parameters and gradient paths being optimized. These are distinct design choices within the same generation process.

3.2 Historical Context and Attention

Historical state selection. For a history window of W chunks, let $\mathcal{R}_{b,j} \subseteq \{1, \ldots, \min(W, b-1)\}$ contain the selected history offsets. A policy $\Phi_b(j, r) \in \{0, \ldots, K\}$ specifies the noise state selected from chunk $b - r$:

$$\mathcal{H}_{b,j}^{\Phi} = \left\{ x_{b-r}^{\Phi_b(j,r)} : r \in \mathcal{R}_{b,j} \right\}. \tag{2}$$

The selection may depend on chunk position and causally available information. For position-independent instances, we use the shorter notation $\phi(j, r)$. Clean, same-level, and progressive history are three representative choices:

$$\phi_{\text{clean}}(j, r) = 0, \qquad \phi_{\text{same}}(j, r) = j, \qquad \phi_{\text{prog}}(j, r) = \max(j - r, 0). \tag{3}$$

They respectively select clean endpoints, states at the current noise level, and states that become cleaner with historical distance. Figure 1 visualizes these representative instances of the general selection rule.

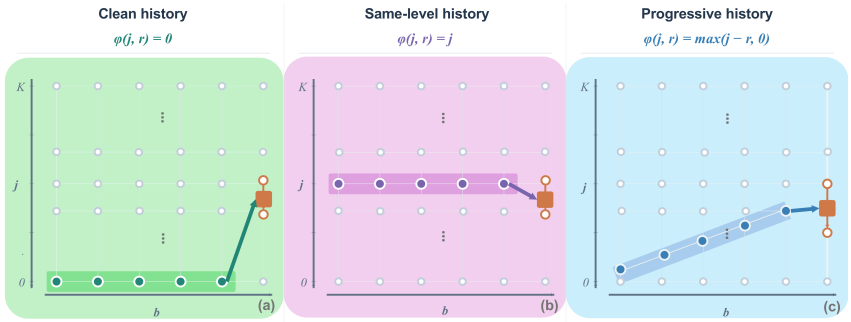

Figure 1: Representative historical context policies. Columns correspond to video chunks and rows to denoising levels. Highlighted historical states condition the current denoising update shown in orange: (a) clean history, (b) same-level history, and (c) progressive history.

Effective context and visibility. $\mathcal{H}^{\Phi}_{b,j}$ specifies historical latent states, whereas $\mathcal{C}_{b,j}$ contains the representations actually used by the denoising network. These representations are computed from the selected latent states, typically stored as KV caches, and read from those caches during denoising. The mask $M_{b,j}$ defines token-level visibility:

$$M_{b,j}(u,v) = \begin{cases} 0, & \text{query token } u \text{ may attend to key token } v, \\ -\infty, & \text{otherwise.} \end{cases} \tag{4}$$

Here u belongs to the current chunk and v to its history or current-chunk features. The mask is added to attention logits before softmax: zero preserves the connection, while $-\infty$ gives it zero attention weight. Thus ϕ selects the noise level of a historical state, while M determines which tokens from the resulting representation each query can access. This notation supports dense causal attention as well as query-dependent sparse access within the causal history. In our chunk-causal setting, future chunks are masked, while within-chunk attention may be bidirectional. Historical features are constructed under the same temporal restriction.

3.3 Fully Serial and Pipelined Execution

Forward inference can proceed strictly one chunk at a time or overlap the denoising of multiple chunks. The key distinction is whether a new chunk must wait for its predecessor to complete the full denoising trajectory.

Fully serial execution. The model performs all K denoising updates for chunk b before starting chunk $b+1$:

$$x_1^K \to \cdots \to x_1^0 \to x_2^K \to \cdots \to x_2^0 \to \cdots . \tag{5}$$

Generating B chunks therefore requires BK sequential chunk-level updates. The first output requires K updates, and successive outputs are separated by another K updates. Apart from stored history, only one chunk is actively denoised, so denoising work does not overlap across chunks.

Pipelined execution. Multiple chunks occupy different denoising stages, allowing their forward computations to be grouped or overlapped. A new chunk may begin before the preceding chunk completes denoising. For a pipeline that admits one new chunk per round and advances every active chunk by one noise level, the active states at the start of round n are

$$\left[x_n^K,\ x_{n-1}^{K-1},\ \ldots,\ x_{n-K+1}^1\right], \tag{6}$$

with chunk indices outside $1,\ldots,B$ omitted during pipeline filling and draining. Under sufficient concurrency, including timely construction of the required context features, this schedule spans $K+B-1$ sequential rounds. It still performs BK chunk-level denoising updates: the reduction is in the number of sequential rounds rather than the number of updates per chunk. The first output traverses K rounds, while a filled pipeline produces one clean chunk per round. Higher concurrency

can improve throughput but requires more active states and intermediate features. Figure 2 contrasts the two execution patterns.

Restrictions on context selection. Causality restricts cross-chunk conditioning to the past. In addition, a historical state selected by the policy must already have been reached, and its features must be available when read. Temporal causality alone does not require clean history; the execution pattern determines how far each historical chunk has progressed and therefore constrains the possible state choices.

Under fully serial execution, earlier chunks have completed their entire denoising trajectories. Clean history is therefore available, and retaining the required intermediate states and representations also permits same-level or progressive history. Fully serial execution is not limited to clean history.

In the pipeline of Eq. (6), when chunk b is at level j, chunk $b-r$ has reached level $\max(j-r,0)$. For states generated by the current rollout, assuming earlier states are retained, latent availability requires

$$\Phi_b(j,r) \geq \max(j-r,0) \quad \text{for every historical offset actually read.} \tag{7}$$

For example, with $K=3$, round two processes x_2^3 alongside x_1^2. Chunk two can select the retained level-3 state of chunk one under the same-level policy, or its current level-2 state under the progressive policy, but it cannot yet select the clean endpoint.

Same-level features can be reused from earlier rounds. Progressive selection may require features constructed in the current round, which can be supplied by a joint causal forward pass or layer-wise execution that respects the same visibility and feature dependencies. Thus both the selected latent states and their effective representations must be available.

3.4 Training Policies, State Sources, and Supervision

Policy alignment. When adapting a model to a specified inference process, the primary context alignment target is

$$\Phi^{\text{train}} = \Phi^{\text{infer}}, \tag{8}$$

together with consistent effective attention visibility and context representation. For example, adaptation to same-level inference uses same-level history during training rather than only clean history. If training covers a broader family of policies, that family should include the target inference policy.

Data-derived and model-generated history. Historical states can be constructed directly from ground-truth data, using either clean or noised tokens (TF and DF; Chen et al., 2024; Huang et al., 2025), or produced through the model's own generation process (SF and Rolling Forcing; Huang et al., 2025; Liu et al., 2025). Between these two constructions, partial denoising of corrupted ground-truth chunks introduces model-generated deviations while retaining an anchor to real data (Resampling Forcing; Guo et al., 2025). The source of history therefore matters: matching historical noise levels alone does not ensure matching state distributions or dependencies on the current chunk.

Supervision objectives. Context construction and supervision are separate design choices. Using the prediction $y_{b,j}$ defined above, common objectives can be summarized in three forms. Local regression uses

$$\mathcal{L}_{\text{local}} = \mathbb{E}\left[\frac{\sum_{b,j} a_{b,j} w_{b,j} \|y_{b,j} - y^\star_{b,j}\|_2^2}{\sum_{b,j} a_{b,j}}\right], \tag{9}$$

where $y^\star_{b,j}$ is the analytical target for the chosen prediction parameterization, $a_{b,j} \in \{0,1\}$ marks supervised nodes, and $w_{b,j}$ is a loss weight. Teacher prediction and trajectory matching use

$$\mathcal{L}_{\text{teacher}} = \mathbb{E}\left[\frac{\sum_{b,j} a_{b,j} w_{b,j} \|y_{b,j} - y^T_{b,j}\|_2^2}{\sum_{b,j} a_{b,j}}\right], \tag{10}$$

$$\mathcal{L}_{\text{trajectory}} = \mathbb{E}\left[\left\| S_j(x_b^j, y_{b,j}; \xi_{b,j}) - x_{b,T}^{j-1} \right\|_2^2\right], \tag{11}$$

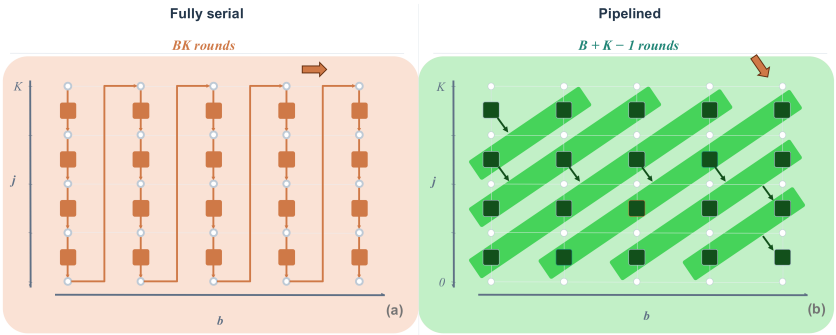

Figure 2: Fully serial and pipelined denoising. (a) Serial execution completes all K updates of one chunk before starting the next, yielding BK sequential rounds. (b) Pipelined execution advances multiple chunks at staggered denoising levels, yielding $B + K - 1$ rounds under sufficient parallel resources while preserving the same BK chunk-level updates.

where $y^T_{b,j}$ is a teacher prediction in the student's parameterization and $x^{j-1}_{b,T}$ is the state obtained by integrating the teacher from x^j_b to t_{j-1}. Finally, video-level distribution matching can be written as

$$\mathcal{L}_{\text{distribution}} = \mathbb{E}_c\left[D\left(p^{\text{rollout}}_\theta(\cdot \mid c), p_{\text{target}}(\cdot \mid c)\right)\right], \tag{12}$$

$$\mathcal{L}_{\text{DMD}} = \mathbb{E}_{c,t}\left[D_{\text{KL}}\left(p^{\text{rollout}}_{\theta,t}(\cdot \mid c) \,\middle\|\, p_{\text{target},t}(\cdot \mid c)\right)\right]. \tag{13}$$

Here $p^{\text{rollout}}_\theta$ is the joint distribution of generated video $x^0_{1:B}$, p_{target} is the data or teacher distribution, and D is a distribution discrepancy. The subscript t denotes applying the same forward noising process to both distributions. The reverse-KL distribution-matching objective does not require paired target samples (Yin et al., 2024b). These objectives can be combined with different state sources at different stages of training.

Training–inference consistency. At the distributional level, training should reflect the joint distribution of current denoising states and historical context encountered during inference. Training on model-generated history exposes the model to its own prediction errors and their propagation across chunks, which can help mitigate error accumulation during autoregressive generation. Training and inference need not use identical batching or execution schedules, provided that conditional dependencies and effective context representations are preserved.

In practice, early training often begins with data-derived history such as TF or DF even though inference uses model-generated history. At initialization, the model's output distribution can be far from the target distribution. Local regression, teacher prediction, or trajectory matching on heavily corrupted model history can then emphasize the undesirable pattern of erroneous history paired with a clean current target, while video-level DMD may provide a weak signal when the two distributions are too far apart. An intuitive analogy is that TF and DF provide instruction along valid solution paths, whereas SF-like training evaluates and corrects the model on its own work. They serve different stages of learning rather than opposing objectives. A streaming video training recipe therefore jointly specifies the context policy, state source, and supervision objective.

In summary, the framework relates model and sampler choices, historical information, attention and context representations, execution, and training. Within this broader design space, our empirical study focuses on historical denoising-state selection through three representative policies, examining generation quality and inference efficiency, complemented by the training-side study of fake-score adaptation.

4 Training

Within the design space developed in Section 3, we fix the backbone, video chunking, and sampling schedule, and compare clean, same-level, and progressive history in generation quality, long-video

behavior, and steady-state inference efficiency. All three policies use rank-64 LoRA adaptation of the fake-score network. Full-parameter fake-score results under clean and progressive history provide additional comparisons of the parameter update scope.

Context-Policy Adaptation. All three policies initialize the student from the causal Wan2.1-T2V-1.3B TF-sCM checkpoint (Team Wan et al., 2025) released by Causal-rCM (Zheng et al., 2026) and use the same text training data and DMD training setup, differing in the target context policy. TF-sCM adapts the continuous-time consistency distillation objective of sCM (Lu & Song, 2025) to teacher forcing, conditioning each chunk on clean ground-truth history. The frozen distillation teacher is the bidirectional Wan2.1-T2V-14B model, and the fake-score network is initialized from the same 14B checkpoint.

During training, the student rolls out from noise under text conditioning, generating and reading history according to the target context policy. Each rollout represents an approximately five-second video at 832×480 resolution and 16 FPS (81 frames). The Wan2.1 VAE applies $4\times$ temporal compression, yielding 21 latent frames organized into seven chunks of three latent frames each. All three policies use a fixed four-step denoising schedule with random early exit: each rollout uniformly selects one of the four predictions as its exit point, and the resulting clean-state predictions form the generated video.

The student is updated with a video-level DMD loss, while the fake-score network is updated with a velocity-prediction regression loss on generated samples. Gradients propagate only through the final denoising prediction of each chunk, whose forward computation is recomputed by replay (Yin et al., 2024b; Huang et al., 2025; Zheng et al., 2026).

All three policies use AdamW with BF16 mixed precision and a global effective batch size of 16. The student and fake-score learning rates are 2×10^{-6} and 4×10^{-7}, respectively, with one student update per five fake-score updates. We maintain an exponential moving average (EMA) of student parameters and use the EMA weights for evaluation. Denoising time points, teacher CFG, DMD time sampling, and the remaining hyperparameters are provided in Appendix A.

Fake-Score Parameterization. The three-policy comparison uses rank-64 fake-score LoRA (Hu et al., 2022), with full-parameter student updates throughout. For all three policies, the pretrained fake-score backbone is frozen, and LoRA branches are added to the linear projections in its self-attention, cross-attention, and feed-forward layers, with alpha set to 64. We additionally conduct full-parameter fake-score adaptation experiments under clean and progressive history.

5 Experimental Results

We compare clean, same-level, and progressive history to assess how historical noise levels affect generation quality. We also benchmark the steady-state inference efficiency of serial and pipelined execution.

We evaluate all 16 VBench dimensions (Huang et al., 2024) with three seeds per prompt. Each video contains 81 frames at 832×480 resolution and 16 FPS (approximately five seconds). Video chunking and training settings follow Section 4; other information is provided in Appendix B. All scores are reported on a 100-point scale, with Total, Quality, and Semantic computed using the evaluation pipeline's normalized, weighted aggregation.

The three-policy comparison uses rank-64 fake-score LoRA, with the training setup in Section 4. We additionally report full-parameter fake-score configurations for clean and progressive history. Clean-history inference includes clean-context refresh. For each adaptation route, we report results from the checkpoint with the highest Total score among those evaluated.

Quantitative Results. Among the LoRA64 configurations in Table 1, same-level history performs comparably to clean history, with a 0.076-point higher Total score, while progressive history improves Total by 0.434 points and achieves the highest aggregate score. Both also score higher in Quality and Semantic. Thus, conditioning on incompletely denoised history does not necessarily compromise generation quality: intermediate states can also support effective streaming generation.

Table 1: Generation quality across context policies and fake-score updates. Trainable parameters refer to the fake-score network. Higher is better.

Context policy	Fake-score update	Trainable params.	Total	Quality	Semantic
Clean: $\phi(j, r) = 0$	Full-parameter	14.288B	84.446	84.862	82.783
	LoRA64	0.307B	85.165	85.493	83.850
Progressive: $\phi(j, r) = \max(j - r, 0)$	Full-parameter	14.288B	84.126	84.477	82.720
	LoRA64	0.307B	**85.599**	**85.961**	**84.149**
Same-level: $\phi(j, r) = j$	LoRA64	0.307B	85.240	85.541	84.038

The complete dimension-level results for these three LoRA64 configurations are provided in Appendix C. Same-level and progressive history score higher in dynamic degree, multiple objects, and imaging quality. Clean history, however, scores higher in subject and background consistency, temporal flickering, motion smoothness, and aesthetic quality. The aggregate gains therefore reflect a trade-off across dimensions rather than a uniform improvement in temporal consistency or visual quality.

Qualitative Results and Limitations. We further compare the visual quality of videos generated under matched conditions. In our qualitative comparisons, clean history tends to produce the most stable backgrounds, but moving subjects often exhibit implausible behavior, such as overlapping limbs or objects interpenetrating their surroundings. Progressive history produces more plausible subject motion; compared with same-level history, it also exhibits fewer abrupt visual changes and more stable backgrounds. Figure 3(a) illustrates vehicle interpenetration in the clean-history result, whereas the other two examples retain a more intact bus structure at the displayed times.

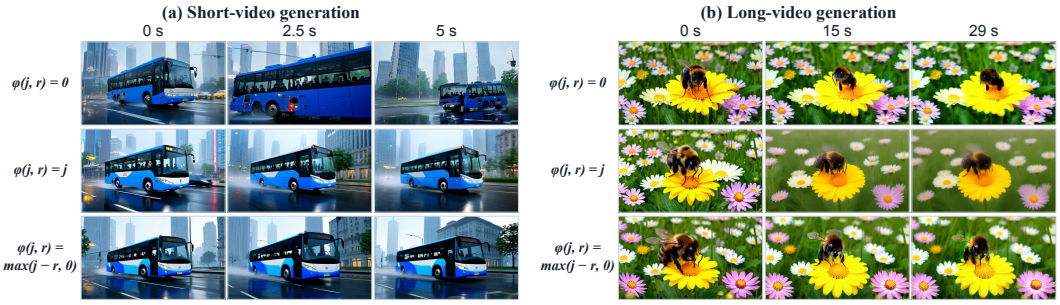

Figure 3: Three-policy comparisons under matched conditions and policy-specific weights. (a) Short-video example for *"a bus accelerating to gain speed"*. (b) Long-video example with prompt excerpt: *"A large, fuzzy bumblebee hovers over a bright yellow daisy, its wings fluttering rapidly."* Columns show the indicated times.

We hypothesize that these differences reflect how strongly generation relies on preceding frames. Fully denoised history provides detailed information, which may encourage the model during training to reproduce previous appearances rather than adapt them to physically plausible motion. Noisy history provides less precise information about past frames, requiring the model to rely more on learned motion and scene priors when generating the current frame. This weaker dependence may support more plausible motion but also reduce background stability, particularly in the initial frames of generation.

Long-video generation. We compare approximately 30-second videos using matched prompts and noise seeds, with each policy's corresponding trained weights. In our qualitative observations, clean-history refresh tends to exhibit subject-shape inconsistency and unnatural motion, while backgrounds often remain nearly frozen, resembling static textures despite their high temporal consistency. Progressive history better preserves subject consistency and supports more plausible subject and background motion. Slight background discontinuities can occur during its initial frames but subside as generation proceeds, possibly because early conditioning is dominated by noisy historical

states before cleaner history accumulates. Figure 3(b) illustrates the contrast in subject morphology: clean-history refresh exhibits inconsistent changes in the bee's body shape, whereas progressive history better preserves its form. In the full videos, clean-history refresh shows little clear foraging-like interaction in the latter half, while progressive history maintains more coherent flower interaction and more natural movement of the surrounding vegetation.

LoRA versus Full-Parameter Fake-Score Adaptation Table 1 also compares LoRA64 and full-parameter fake-score adaptation under clean and progressive history. The student remains fully trainable in all configurations. LoRA64 updates approximately 2.15% as many fake-score parameters, while improving Total by 0.719 and 1.473 points under clean and progressive history, respectively, with gains in both Quality and Semantic. Greater adaptation freedom in the fake-score network does not necessarily yield better student generation quality in these comparisons. We hypothesize that low-rank updates impose a structural constraint on adaptation while leveraging pretrained denoising capabilities. This constraint may offer a more favorable balance between adaptation and stability when tracking the evolving student distribution, potentially providing more effective distillation signals to the student.

Inference Efficiency. We benchmark clean-history serial execution and progressive-history pipelined execution on 4, 8, and 12 Ascend 910B2C devices using context parallelism (CP). Each run generates 477 frames at 832×480 resolution and 16 FPS, with four denoising steps and one clean-state KV refresh per chunk. We measure 20 steady-state output chunks and report the median over three trials, including communication and cache operations but excluding VAE decoding and pipeline fill and drain.

Table 2: Steady-state DiT latency (ms/chunk) and speedup.

Devices (CP)	Clean serial (ms/chunk)	Progressive pipeline (ms/chunk)	Speedup
4	527.7	336.0	$1.57\times$
8	441.7	222.4	$1.99\times$
12	626.3	221.0	$2.83\times$

Progressive history combines multiple denoising nodes in one forward pass, achieving 1.57–$2.83\times$ steady-state speedups under the evaluated configurations. As CP reduces the per-device workload, this grouping can help amortize execution and communication overheads.

6 Conclusion

We studied streaming video diffusion as a joint design problem over historical context, execution schedule, and training. Our framework separates four elements that are coupled in existing systems: the policy Φ selecting the denoising state of historical chunks, the attention mask M governing visibility, the source of historical states, and the supervision objective. This separation makes the comparison of context policies well defined, since it identifies exactly what must be held fixed to attribute an observed difference to Φ alone.

Within this framework we compared clean, same-level, and progressive history under a fixed backbone, training data, chunking, denoising schedule, and DMD objective. With rank-64 fake-score LoRA, both non-clean policies exceed clean history in aggregate VBench score (85.24 and 85.60 versus 85.16), with progressive history additionally showing more coherent motion and better subject consistency in long-video comparisons. Fully denoised history is therefore not a prerequisite for high-quality streaming generation. Because progressive history removes the dependency that forces a chunk to wait for its predecessor to finish denoising, it admits pipelined execution, which reduces the number of communication rounds per output chunk and yields 1.57–$2.83\times$ steady-state DiT speedups under our evaluated configurations. On the training side, LoRA adaptation of the DMD fake-score network improves generation quality while updating 2.15% as many parameters as full-parameter adaptation. Together these results argue for treating historical conditioning, execution scheduling, and training as one design space rather than three independent choices.

AI USE STATEMENT

We used a large language model as a general-purpose writing assistant for grammatical correction and for improving the clarity of exposition in the manuscript. All research ideas, the analytical framework, the experimental design, the implementation, and the analysis of results are the authors' own. The authors have verified all technical claims and take full responsibility for the content of this paper.

REPRODUCIBILITY STATEMENT

Section 4 and Appendix A describe model initialization, training objectives, and optimization settings. Section 5 specifies generation settings and inference timing protocols. Appendix B details the evaluation procedure and checkpoint selection, while Appendix C provides results for all 16 VBench dimensions.

ETHICS STATEMENT

This work investigates context selection, execution, and training for streaming video generation. Like other video generation methods, it may inherit biases from pretrained models and can be misused to create deceptive or non-consensual media. Generated videos should be clearly identified as synthetic rather than presented as factual recordings. Responsible use should respect privacy, consent, and the licenses and usage restrictions of the underlying models and data.

REFERENCES

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*, 2023a. URL https://arxiv.org/abs/2311.15127.

Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22563–22575, 2023b. URL https://arxiv.org/abs/2304.08818.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion Forcing: Next-token prediction meets full-sequence diffusion. In *Advances in Neural Information Processing Systems*, volume 37, pp. 24081–24125, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2aee1c4159e48407d68fe16ae8e6e49e-Abstract-Conference.html.

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. URL https://arxiv.org/abs/2504.13074.

Tianrui Feng, Zhi Li, Shuo Yang, Haocheng Xi, Muyang Li, Xiuyu Li, Lvmin Zhang, Keting Yang, Kelly Peng, Song Han, Maneesh Agrawala, Kurt Keutzer, Akio Kodaira, and Chenfeng Xu. StreamDiffusionV2: A streaming system for dynamic and interactive video generation. *Proceedings of Machine Learning and Systems*, 8:1656–1669, 2026. URL https://streamdiffusionv2.github.io/.

Yuwei Guo, Ceyuan Yang, Hao He, Yang Zhao, Meng Wei, Zhenheng Yang, Weilin Huang, and Dahua Lin. End-to-end training for autoregressive video diffusion via self-resampling. *arXiv preprint arXiv:2512.15702*, 2025. URL https://arxiv.org/abs/2512.15702.

Roberto Henschel, Levon Khachatryan, Hayk Poghosyan, Daniil Hayrapetyan, Vahram Tadevosyan, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. StreamingT2V: Consistent, dynamic, and extendable long video generation from text. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2568–2577, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Henschel_StreamingT2V_Consistent_Dynamic_and_Extendable_Long_Video_Generation_from_Text_CVPR_2025_paper.html.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 8633–8646, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/39235c56aef13fb05a6adc95eb9d8d66-Abstract-Conference.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self Forcing: Bridging the train-test gap in autoregressive video diffusion. In *Advances in Neural Information Processing Systems*, volume 38, pp. 167283–167308, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/f4823f831af67a3ef15e41a85434422a-Abstract-Conference.html.

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

models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21807–21818, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Huang_VBench_Comprehensive_Benchmark_Suite_for_Video_Generative_Models_CVPR_2024_paper.html.

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*, volume 37, pp. 89834–89868, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a397986e0f34d4b1f0b640686ceaeff7-Abstract.html.

Dan Kondratyuk, Lijun Yu, Xiuye Gu, Jose Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, Krishna Somandepalli, Hassan Akbari, Yair Alon, Yong Cheng, Joshua V. Dillon, Agrim Gupta, Meera Hahn, Anja Hauth, David Hendon, Alonso Martinez, David Minnen, Mikhail Sirotenko, Kihyuk Sohn, Xuan Yang, Hartwig Adam, Ming-Hsuan Yang, Irfan Essa, Huisheng Wang, David A. Ross, Bryan Seybold, and Lu Jiang. VideoPoet: A large language model for zero-shot video generation. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 25105–25124. PMLR, 2024. URL https://proceedings.mlr.press/v235/kondratyuk24a.html.

Kunhao Liu, Wenbo Hu, Jiale Xu, Ying Shan, and Shijian Lu. Rolling Forcing: Autoregressive long video diffusion in real time. *arXiv preprint arXiv:2509.25161*, 2025. URL https://arxiv.org/abs/2509.25161.

Cheng Lu and Yang Song. Simplifying, stabilizing and scaling continuous-time consistency models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=LyJi5ugyJx.

Simian Luo, Yiqin Tan, Suraj Patil, Daniel Gu, Patrick von Platen, Apolinário Passos, Longbo Huang, Jian Li, and Hang Zhao. LCM-LoRA: A universal stable-diffusion acceleration module. *arXiv preprint arXiv:2311.05556*, 2023. URL https://arxiv.org/abs/2311.05556.

David Ruhe, Jonathan Heek, Tim Salimans, and Emiel Hoogeboom. Rolling diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 42818–42835. PMLR, 2024. URL https://proceedings.mlr.press/v235/ruhe24a.html.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *The Tenth International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TIdIXIpzhoI.

Kiwhan Song, Boyuan Chen, Max Simchowitz, Yilun Du, Russ Tedrake, and Vincent Sitzmann. History-guided video diffusion. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 56242–56280. PMLR, 2025. URL https://proceedings.mlr.press/v267/song25b.html.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252. PMLR, 2023. URL https://proceedings.mlr.press/v202/song23a.html.

Team Wan, Ang Wang, Baole Ai, Bin Wen, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. URL https://arxiv.org/abs/2503.20314.

Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. VideoGPT: Video generation using VQ-VAE and transformers. *arXiv preprint arXiv:2104.10157*, 2021. URL https://arxiv.org/abs/2104.10157.

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 *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ce31378e9f41d8907e97dab172b6c559-Abstract-Conference.html.

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 *Advances in Neural Information Processing Systems*, volume 37, pp. 47455–47487, 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/54dcf25318f9de5a7a01f0a4125c541e-Abstract-Conference.html.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6613–6623, 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/papers/Yin_One-step_Diffusion_with_Distribution_Matching_Distillation_CVPR_2024_paper.pdf.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22963–22974, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yin_From_Slow_Bidirectional_to_Fast_Autoregressive_Video_Diffusion_Models_CVPR_2025_paper.html.

Kaiwen Zheng, Guande He, Min Zhao, Jintao Zhang, Huayu Chen, Jianfei Chen, Chen-Hsuan Lin, Ming-Yu Liu, Jun Zhu, and Qianli Ma. Causal-rCM: A unified teacher-forcing and self-forcing open recipe for autoregressive diffusion distillation in streaming video generation and interactive world models. *arXiv preprint arXiv:2606.25473*, 2026. URL https://arxiv.org/abs/2606.25473.

A TRAINING CONFIGURATIONS

Table 3 summarizes the configurations used to adapt the TF-sCM-initialized student to clean, same-level, and progressive history, together with the additional full-parameter fake-score variant under progressive history.

Table 3: Shared training configurations for clean, same-level, and progressive context adaptation with fake-score LoRA. The additional full-parameter fake-score variant uses progressive history; the clean-history full-parameter result is retained as a separate reference. The student is fully trainable in all configurations.

Item	Setting
Student initialization	Causal-rCM; Wan2.1-T2V-1.3B; TF-sCM; `c3-3`.
Teacher / fake-score backbone	Wan2.1-T2V-14B; frozen teacher.
Training samples	Student-generated videos; 81 frames at 16 FPS, approximately five seconds.
VAE temporal compression	Wan2.1 VAE; $4\times$ temporal compression; 81 video frames correspond to 21 latent frames.
Video chunking	Three latent frames per chunk; seven chunks.
Spatial resolution	832×480.
Denoising schedule / random exit	Input times $1,\ 15/16,\ 5/6,\ 5/8$; uniformly sampled exit at prediction 1–4.
DMD supervision	The complete generated sequence.
Teacher CFG / DMD time sampling	5 / UniformShift with shift 5.
Gradient scope / backward computation	Stop-gradient historical caches; replay the final denoising prediction of each chunk.
Optimizer implementation	NPU Master AdamW (`npumasteradamw`).
Student / fake-score learning rate	2×10^{-6} / 4×10^{-7}.
AdamW parameters	Betas $(0, 0.999)$; weight decay 0.01; epsilon 10^{-8}.
Student : fake-score updates	$1 : 5$.
Microbatch / gradient accumulation / effective batch size	1 per data-parallel replica / 1 / 16 globally.
Training hardware and parallelism	16 Ascend 910B2C devices; data-parallel degree 16; context-, tensor-, and pipeline-parallel degrees 1.
Precision / training seed	BF16 / 0.
EMA	Causal-rCM's iteration-dependent EMA rule with `ema.rate=0.1`.
Evaluation weights	Student `net_ema`, exported in BF16.
LoRA parameters	Rank 64; alpha 64; dropout 0; zero-output initialization.
LoRA target layers	Q, K, V, and output projections in self- and cross-attention; both feed-forward linear projections.
LoRA / full-parameter update scope	LoRA: fake-score adapters only. Full-parameter: all fake-score parameters, without LoRA. Both: all student parameters.

B EVALUATION PROTOCOL AND CHECKPOINT SELECTION

Evaluation coverage. We evaluate all 16 dimensions on the VBench evaluation set, with 720 videos per configuration. All models use the same prompts, seed indices, and sampling configuration. Videos contain 81 frames at 16 FPS, with 832×480 resolution, four denoising steps, and chunks of three latent frames.

Score aggregation. Following the existing evaluation pipeline, each raw dimension score is normalized by its prescribed range and clipped to $[0, 1]$. Quality dimensions have unit weight except dynamic degree, whose weight is 0.5, giving a total group weight of 6.5. The nine semantic dimensions have unit weight. The total score is 0.8 times the quality score plus 0.2 times the semantic

score. Tables 1 and 4 multiply their respective aggregate and raw scores by 100. The raw dimension scores in Table 4 are reported before the normalization used for aggregation.

Quantitative checkpoints. The clean-history LoRA64 model is trained under the clean-history policy using the same student initialization, text data, and DMD training setup as the other two policies, with rank-64 fake-score LoRA and full-parameter student updates. It uses four-step sampling with clean-context refresh. For Table 1, the selected clean, same-level, and progressive fake-score LoRA checkpoints are 1800, 2250, and 3150, respectively. They maximize the aggregate score among each route's completed evaluations under the above protocol. Checkpoint numbers denote stored iteration identifiers; equal iteration counts are not imposed across routes. The clean-history full-parameter reference and the progressive full-parameter checkpoint at iteration 2500 are also reported in that table.

C Complete Dimension-Level Results

Table 4 reports all 16 VBench dimensions for the three LoRA64 model configurations in Table 1, under the evaluation protocol in Appendix B.

Table 4: All 16 VBench dimensions for the three LoRA64 context policies on the VBench evaluation set. Values are raw scores multiplied by 100, before normalization for aggregation. Higher is better in each dimension; the best result in each row is bold, including ties.

Dimension	Clean	Same-level	Progressive
Quality dimensions			
Subject consistency	**96.469**	95.801	96.286
Background consistency	**95.448**	95.212	95.406
Temporal flickering	**98.595**	98.424	98.453
Motion smoothness	**98.408**	97.886	98.089
Aesthetic quality	**66.721**	66.352	66.082
Imaging quality	68.788	69.347	**70.049**
Dynamic degree	77.778	84.722	**86.111**
Semantic dimensions			
Object class	94.167	93.646	**96.979**
Multiple objects	86.111	86.458	**87.153**
Human action	**100.000**	**100.000**	**100.000**
Color	85.271	86.660	**89.688**
Spatial relationship	80.945	**81.961**	81.504
Scene	**70.312**	68.576	64.670
Appearance style	**22.102**	21.846	21.936
Temporal style	26.412	**26.903**	26.543
Overall consistency	26.476	**26.884**	26.822