OPSD-V

On-Policy Self-Distillation for Post-Training Few-Step Autoregressive Video Generators

Hongyu Liu[1,2], Chun Wang[1,2], Feng Gao[1,†], Xuanhua He[1,2],
Yue Ma[2], Ziyu Wan[3], Yong Zhang[1,‡], Xiaoming Wei[1], Qifeng Chen[2,†]
[1]Meituan [2]HKUST [3]City University of Hong Kong
[†]Corresponding authors [‡]Project lead
hliudq@connect.ust.hk

We propose OPSD-V, an on-policy self-distillation paradigm for post-training few-step autoregressive (AR) video diffusion models. Existing few-step AR video generators, often obtained through DMD-style distillation, can generate long videos with low latency, but still suffer from error accumulation and weakened motion dynamics during long autoregressive rollout. OPSD-V aims to further reduce long-horizon degradation and improve motion dynamics while preserving the original few-step inference path. Our key idea is to introduce real long-video data as temporal context during training and use it to provide dense trajectory-level supervision. Compared with relying only on a short-clip teacher distribution, real long videos offer a richer and cleaner target distribution for supervising long AR rollouts. Specifically, the student follows the exact inference-time rollout, generating each chunk conditioned on its own previously generated KV cache. In parallel, the teacher is evaluated at the same student-visited denoising states, but uses a cleaner AR-consistent temporal cache in which older history can be replaced by real-video context. To maintain autoregressive consistency and prevent the teacher from becoming a fully teacher-forced oracle, both branches share an initial real-video prefix, and the teacher keeps its most recent cache chunk generated by the model itself. This design provides dense denoising-level corrective targets under on-policy AR cache dynamics, without changing the sampler, number of denoising steps, or inference-time cache mechanism. We apply OPSD-V to representative few-step AR video models, including Self-Forcing and LongLive. Experiments show consistent improvements in visual quality, motion dynamics, and VBenchLong scores. In a user study with 10 participants comparing 20 video pairs, OPSD-V is preferred over the base models in 66.0% of overall-preference judgments (82.5% excluding ties), demonstrating the effectiveness of on-policy self-distillation with real long-video context for long-horizon AR video generation.

Project page: https://meigen-ai.github.io/OPSD-V
Code: MeiGen-AI/OPSD-V

1 Introduction

Video generation has rapidly evolved from short, offline clip synthesis to large-scale video foundation models capable of producing high-resolution, text-aligned, and temporally coherent videos. Modern video generation models are largely built upon diffusion transformers (DiTs) (Peebles & Xie, 2022), whose scalability has enabled substantial progress in visual fidelity, motion quality, prompt following, and long-video generation (Yang et al., 2024; Kong et al., 2024; Wan Team, 2025; Meituan LongCat Team et al., 2025; HaCohen et al., 2026; Brooks et al., 2024). Despite these advances, most large-scale video foundation models are still primarily designed for offline generation, where an entire video is synthesized after sampling rather than produced continuously during user interaction.

In contrast to offline video generation, real-time video generation requires a model to produce content sequentially with low viewing latency. This requirement naturally favors autoregressive (AR) video models, where each new frame or chunk is generated conditioned on previous outputs, and transformer KV caches are reused to maintain historical context efficiently. To make AR generation practical, recent methods combine causal video modeling with few-step distillation, which compresses slow diffusion or flow models into efficient few-step generators (Yin et al., 2024b;a; Wang et al., 2023). Building on this recipe, CausVid distills a bidirectional video DiT into a causal AR student for fast streaming generation (Yin et al., 2025), while Self-Forcing further aligns training with inference by rolling out the model on its own generated frames and rolling KV cache (Huang et al., 2025a). Together, these AR and few-step generation techniques are

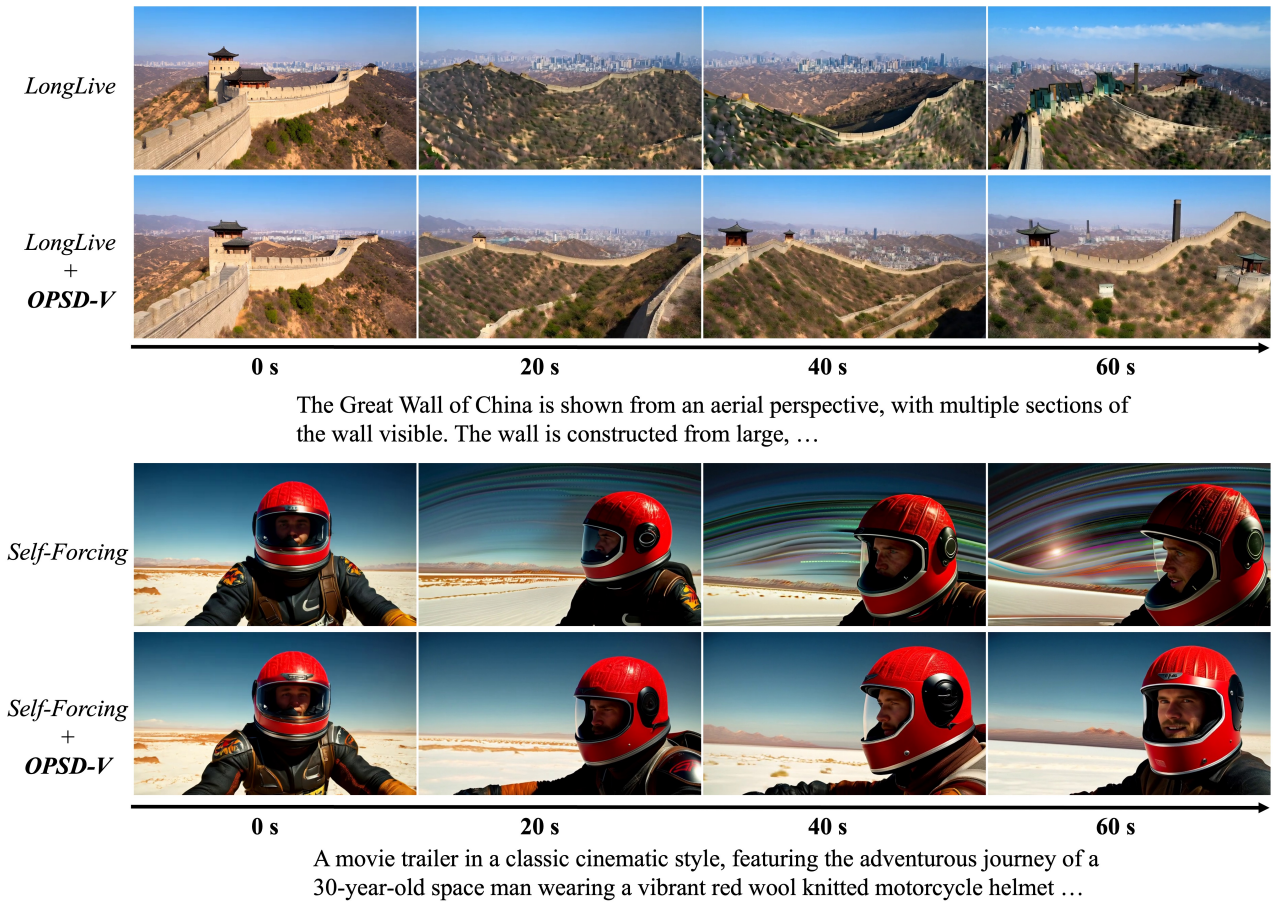

Figure 1: **OPSD-V improves long-horizon autoregressive video generation.** We compare the original base model and its OPSD-V post-trained version under the same prompt and seed. The first example is based on LongLive, and the second example is based on Self-Forcing. In each example, the first row shows the base model result, while the second row shows the result after applying OPSD-V. Our method produces more dynamic and temporally stable videos while reducing long-horizon error accumulation. For LongLive, OPSD-V noticeably reduces artifacts in the grass region and distant background. For Self-Forcing, OPSD-V improves the character's head motion and suppresses background artifacts.

enabling applications such as streaming avatars (Yang et al., 2025a; Team et al., 2026a; Wang et al., 2026), interactive media (Shin et al., 2025; ai et al., 2025), world-model-like applications (Ball et al., 2025; Team et al., 2026b), and embodied intelligence (Li et al., 2026b; Ye et al., 2026), where models must produce videos continuously while maintaining stable identities, motions, and scene structures over long horizons.

Despite the success of combining self-forcing style training with DMD-style few-step distillation, existing AR video generators still suffer from long-horizon degradation. One fundamental limitation is that the target distribution used in DMD-style training is usually provided by a bidirectional short-clip video teacher, which cannot directly supervise truly long autoregressive trajectories. Although recent methods Yang et al. (2025b); Cui et al. (2025); Zhang et al. (2026); Li et al. (2026a) extend rollout length and adopt attention-sink mechanisms Xiao et al. (2023) to mitigate error accumulation, their supervision is still limited by the capability and temporal range of the underlying clip-level teacher. Moreover, DMD itself may introduce side effects such as weakened dynamics and color drift, and its supervision is typically defined at the chunk or clip distribution level rather than as dense corrective targets along the entire denoising trajectory.

This limitation points to an inherent ceiling of existing DMD-style post-training: the few-step sampling path is preserved, but the score-matching signal is still tied to the temporal range and quality of a finite short-clip teacher. This raises a central question for post-training few-step AR video generators:

> *Can real long-video data serve as stronger training supervision while preserving the original few-step AR generation capability?*

Our answer is to use real long videos not as direct teacher-forcing targets, but as privileged temporal context for constructing a cleaner teacher distribution on the student's own rollout states. On-policy self-distillation (OPSD) provides a

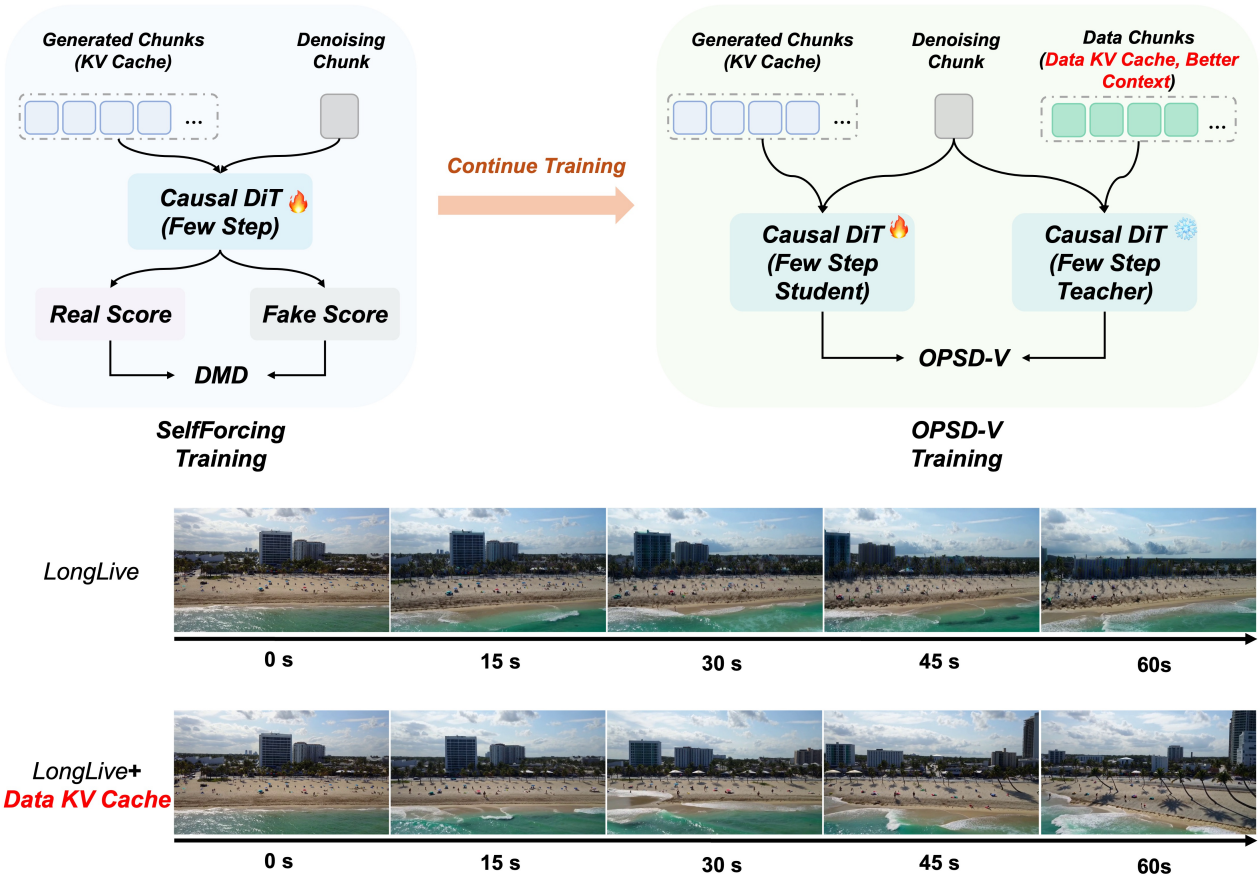

Figure 2: **Overview and motivating cache intervention.** *Top:* Unlike Self-Forcing training, which applies DMD supervision after self-rollout, OPSD-V continues post-training an existing few-step AR generator by aligning an on-policy student with a teacher conditioned on cleaner long-video context. The student writes its own generated chunks into the KV cache, while the teacher is evaluated at the same denoising states using a data-assisted cache. *Bottom:* Before any OPSD-V post-training, we isolate the effect of cache quality using the original LongLive model at test time. Both rows are initialized with the same real first chunk from the source video. The first row is the standard LongLive rollout; in the second row, older KV-cache entries are replaced by those computed from the corresponding real-video chunks, while the most recent cache chunk remains generated by the model. This data-context intervention already improves long-horizon stability, motivating our use of real-video context to construct cleaner teacher targets during training.

suitable framework for this goal. Recently studied in autoregressive large language models (Zhao et al., 2026; Shenfeld et al., 2026; Hübotter et al., 2026), OPSD adopts an on-policy learning paradigm in which the model samples from its current policy as a student, while a stronger teacher distribution is obtained by conditioning the same model on richer in-context information. This formulation naturally provides dense supervision along the model's own trajectory. In our AR video setting, real long videos can serve as such richer context: by conditioning the same video generator on cleaner long-video states, the teacher can provide a more reliable target distribution for the student's rollout trajectory. A closely related work, D-OPSD (Jiang et al., 2026), has demonstrated the feasibility of this context-enhanced self-distillation idea for text-to-image generation, where a real image paired with the text prompt is used as additional context to obtain a stronger teacher distribution. Inspired by this paradigm, we explore using real long videos as reliable temporal context to construct a cleaner teacher distribution, enabling dense denoising-level supervision on the student's autoregressive rollout states.

In this paper, we propose OPSD-V, a cache-aware on-policy self-distillation framework for post-training few-step AR video diffusion models. As illustrated in Fig. 2, OPSD-V starts from an existing few-step AR generator and continues training it with supervision tailored to long autoregressive rollout. A key challenge is that directly using real long videos is non-trivial: even under the same prompt, the student rollout may produce a trajectory different from the real video, so naively using real-video chunks as reconstruction targets or teacher context can introduce semantic and temporal mismatch. Therefore, we treat real videos as privileged temporal context rather than direct output targets. Specifically, the student remains fully on-policy: it denoises each chunk with the original few-step sampler, writes its own generated chunks into the KV cache, and continues from the resulting self-induced temporal states. The teacher is evaluated at the same temporal positions, timesteps, and student-visited noisy latents, but uses a cleaner cache constructed from real

long-video context. Both branches share the same initial real-video prefix to anchor the scene; for each target chunk, the teacher replaces older generated cache history with the corresponding real-video context while retaining the most recent generated chunk. This design reduces accumulated degradation in the teacher context while preserving autoregressive continuation from the student's current state. Dense velocity matching along the fixed few-step denoising trajectory then improves long-horizon generation without changing the original low-latency sampling path.

The lower part of Fig. 2 shows the diagnostic observation that led to this design. Before applying any OPSD-V post-training, we take the original LongLive model and run it at test time from the same real first chunk under two cache settings. In both settings, this first chunk is taken from the source video and used only to initialize the cache. The first setting is standard inference, where the history cache is built entirely from generated chunks. In the second setting, we provide data context only to the older cache history by replacing those KV entries with features computed from the corresponding real-video chunks, while keeping the most recent cache chunk generated by the model itself. No model parameters are updated, and the denoising sampler is unchanged. The fact that this test-time data-context intervention alone produces a visibly more stable long-horizon rollout directly suggests that degradation in the generated KV cache is a key bottleneck. This observation motivates OPSD-V: during training, we use real long-video context as privileged teacher information to provide cleaner targets, while the student still follows the same self-generated cache trajectory it will encounter at deployment.

We perform OPSD-V post-training on a small customized long-video dataset containing 3,800 videos, each approximately one minute in length. To verify the generality of our post-training framework, we apply OPSD-V to two representative few-step AR video generators, Self-Forcing Huang et al. (2025a) and LongLive Yang et al. (2025b), using LoRA-based continued training for both models. **Self-Forcing is a widely compared few-step AR baseline**, but its original training is mainly performed on short clips; using it as the only long-video baseline would make the evaluation incomplete and could overstate gains from long-horizon post-training. Since OPSD-V adopts an attention-sink cache mechanism by default for long-video training and inference, we additionally equip Self-Forcing with the same attention-sink mechanism in all long-video evaluations to make the comparison stronger. **To further ensure fairness and validate OPSD-V on a backbone already trained for streaming long-video generation**, we also train and compare on LongLive, which incorporates streaming long tuning for long-video synthesis. After post-training, OPSD-V consistently improves motion dynamics and effectively mitigates error accumulation during long-horizon generation. Fig. 1 further shows that our method reduces long-rollout degradation while preserving the original few-step AR inference path.

2 Related Work

Video diffusion foundation models. Recent progress in video generation has been largely built upon diffusion and flow-matching generative paradigms, which synthesize data through iterative denoising or continuous transport from noise to data (Song et al., 2020; Lipman et al., 2023; Ho et al., 2020). Combined with scalable diffusion transformer architectures Peebles & Xie (2022), these techniques have substantially improved text-to-video fidelity, motion quality, prompt following, and temporal coherence. Representative systems such as CogVideoX, HunyuanVideo, Wan, and LongCat-Video further scale video diffusion with stronger transformer backbones, latent video autoencoders, and large training corpora (Yang et al., 2024; Kong et al., 2024; Wan Team, 2025; Meituan LongCat Team et al., 2025; Brooks et al., 2024; Zheng et al., 2024). These models typically denoise a temporal window with bidirectional attention, allowing rich interactions among frames and leading to strong offline generation quality. However, their full-window denoising interface and multi-step sampling procedure are less suitable for real-time or interactive scenarios, where frames or chunks need to be emitted sequentially before the entire video is synthesized. This motivates causal autoregressive video generation, which preserves the generative strength of diffusion models while adapting them to streaming inference.

Autoregressive video generation and forcing-based training. Autoregressive (AR) video models generate videos sequentially by conditioning each new frame or chunk on previously generated content, often using transformer KV caches to efficiently reuse historical context. Early causal video diffusion methods commonly follow teacher-forcing or diffusion-forcing paradigms, where the model learns to denoise future frames conditioned on clean or partially noised context frames (Jin et al., 2025; Chen et al., 2024; 2025). To make AR diffusion practical for real-time generation, CausVid distills a bidirectional video DiT into a causal student with DMD-style few-step distillation (Yin et al., 2025; 2023). Self-Forcing further reduces exposure bias by performing autoregressive self-rollout during training, allowing the model to condition on its own generated frames and rolling KV cache (Huang et al., 2025a). Following this direction, Self-Forcing++ (Cui et al., 2025) extends self-rollout to minute-scale horizons; Causal Forcing improves causal AR distillation through a better initialization strategy (Zhu et al., 2026); Reward Forcing introduces reward feedback and EMA attention sinks to improve optimization and inference performance (Zhang et al., 2026); and LongLive develops chunk-level AR generation with KV recache and streaming long tuning for interactive long videos (Yang et al., 2025b). Other works further improve cache management, long-context extrapolation, attention sinks, or inference-time stabilization for open-ended AR generation (Li et al., 2026a; Liu et al., 2025b; Mao et al., 2026; Yi et al., 2025; Yesiltepe et al., 2026). These methods improve the efficiency and stability of causal video generation, but their supervision is still primarily based on short-clip teachers, rollout-level objectives, reward signals, or cache heuristics. More recently, Cai et al. (2026) also explores real long-video data as supervision by introducing a two-head DiT design; however, it targets a non-causal long-video training paradigm

rather than cache-aware post-training for causal few-step AR video generation. In contrast, OPSD-V focuses on post-training an already efficient few-step AR video model by providing dense denoising-level supervision on the student's own inference-time cache states.

On-policy self-distillation. On-policy self-distillation (OPSD) has recently emerged as a way to improve models on the trajectories they actually visit. In autoregressive large language models, the same model can act as both student and teacher under different contexts: the student samples from its current policy, while the teacher distribution is obtained by conditioning the model on richer in-context information (Zhao et al., 2026; Shenfeld et al., 2026; Hübotter et al., 2026; He et al., 2026). This idea has also been extended to step-distilled diffusion and flow models. D-OPSD applies OPSD to text-to-image generation by conditioning the teacher on both the text prompt and its paired real image, enabling supervised tuning along the student's own few-step rollouts without sacrificing few-step inference capability (Jiang et al., 2026). Related OPD-style methods further study trajectory-level distillation for diffusion and flow models (Fang et al., 2026; Li et al., 2026c; Gu et al., 2026). OPSD-V follows this context-enhanced self-distillation principle, but extends it to the AR video setting where the on-policy state is not only the noisy latent and denoising timestep, but also the evolving KV cache written by previously generated chunks. By using real long videos as privileged temporal context, OPSD-V constructs a cleaner teacher distribution and provides cache-aware dense supervision for long-horizon few-step AR video generation.

3 Preliminaries

3.1 Few-Step Autoregressive Video Generation

We consider a causal autoregressive (AR) video diffusion model that generates a video as a sequence of latent chunks. Let $x_{1:N} = \{x_1, \dots, x_N\}$ denote a video divided into N chunks, where each chunk may contain one or more latent frames, and let c denote the text prompt or other conditioning signal. An AR video generator factorizes the video distribution as

$$p_\theta(x_{1:N} \mid c) = \prod_{i=1}^{N} p_\theta(x_i \mid x_{<i}, c).$$

In causal video diffusion transformers, the historical context $x_{<i}$ is implemented through transformer key-value (KV) caches, so previous chunks do not need to be recomputed from scratch at every generation step. We denote the cache state available before generating chunk i as h_i. During inference, the model generates chunk i conditioned on h_i, and then appends the generated chunk into the cache for future chunks.

For a few-step AR video generation model, each chunk is generated by a small number of denoising steps. Starting from Gaussian noise $z_{i,K}$, the model denoises it over a fixed timestep schedule $\{t_K, \dots, t_1\}$. Let f_θ denote the causal video DiT velocity predictor. At denoising step k, the model predicts a velocity field

$$\hat{v}_{i,k} = f_\theta(z_{i,k}, t_k, c, h_i),$$

where $z_{i,k}$ is the current noisy latent of chunk i, t_k is the denoising timestep, and h_i is the historical KV-cache state. The predicted velocity $\hat{v}_{i,k}$ is not the next latent itself; instead, it specifies the denoising direction used by the numerical solver. Given this velocity, a solver transition Φ updates the latent state:

$$z_{i,k-1} = \Phi(z_{i,k}, t_k, t_{k-1}, \hat{v}_{i,k}), \qquad k = K, \dots, 1.$$

After the final denoising step, the clean latent chunk is obtained as

$$\hat{x}_i = z_{i,0}.$$

The generated chunk is then written into the KV cache. We use $\mathrm{KV}_\theta(\hat{x}_i)$ to denote the KV entries obtained from the generated clean chunk, and use $\oplus$ to denote appending new KV entries to the existing cache. The cache update can be written as

$$h_{i+1} = h_i \oplus \mathrm{KV}_\theta(\hat{x}_i).$$

Thus, future chunks depend recursively on previously generated chunks through the evolving KV cache.

Few-step AR video generators are commonly obtained by combining causal video modeling with distribution matching distillation (DMD) or related few-step distillation objectives (Yin et al., 2023; Liu et al., 2025a). DMD compresses a strong but slow diffusion model into an efficient one-step or few-step generator by matching the student distribution to a teacher or data distribution. In AR video generation, methods such as CausVid use DMD-style distillation to adapt a bidirectional video diffusion teacher into a causal few-step student (Yin et al., 2025), while Self-Forcing further performs autoregressive self-rollout during training so that the model conditions on its own generated frames and rolling KV cache (Huang et al., 2025a). These techniques provide the efficient few-step AR generators that serve as the starting point for our post-training framework.

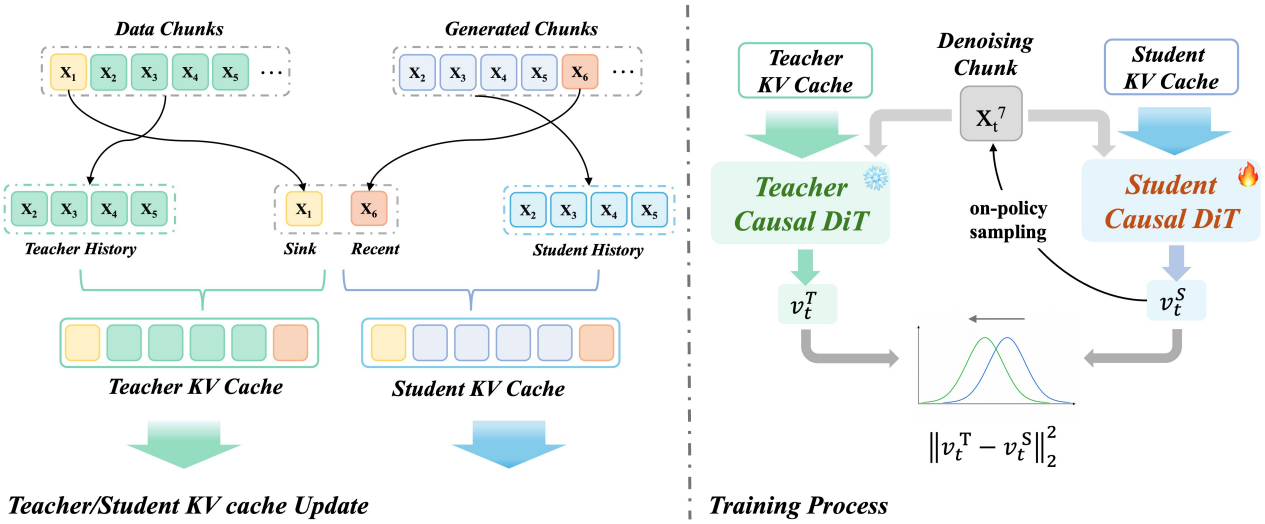

Figure 3: OPSD-V post-trains a few-step autoregressive video generator with cache-aware on-policy self-distillation. The student follows the exact inference-time rollout and writes its own generated chunks into the KV cache. The teacher is evaluated at the same student-visited denoising states, but uses an AR-consistent real-video cache, where older history is replaced by real-video chunks while the most recent chunk remains student-generated.

3.2 On-Policy Self-Distillation

On-policy self-distillation (OPSD) has recently been studied in autoregressive language models as a way to improve a model on the trajectories it actually samples Zhao et al. (2026); Shenfeld et al. (2026). Given an input query q, the model first acts as a student and samples an output trajectory from its current policy:

$$\hat{o}_{1:T} \sim \pi_\theta(\cdot \mid q), \tag{1}$$

where $\hat{o}_{1:T}$ denotes the sampled sequence. The same model, or an exponential-moving-average copy of it, then acts as a teacher under a stronger context. Let r denote additional in-context information, such as demonstrations, intermediate reasoning, or other privileged supervision. At each token position m, the student predicts the next-token distribution conditioned on the query and its own sampled prefix $\hat{o}_{<m}$, while the teacher predicts under the same sampled prefix but with the additional context r:

$$\pi_\theta(\cdot \mid q, \hat{o}_{<m}), \qquad \pi_{\bar{\theta}}(\cdot \mid q, r, \hat{o}_{<m}), \tag{2}$$

where $\bar{\theta}$ denotes the teacher parameters. OPSD then optimizes the student by matching these two distributions along the student-sampled trajectory:

$$\mathcal{L}_{\text{OPSD}} = \mathbb{E}_{\hat{o}_{1:T}\sim\pi_\theta(\cdot|q)}\left[\sum_{m=1}^{T} D\left(\pi_{\bar{\theta}}(\cdot \mid q, r, \hat{o}_{<m}) \,\|\, \pi_\theta(\cdot \mid q, \hat{o}_{<m})\right)\right], \tag{3}$$

where $D(\cdot\|\cdot)$ denotes a distillation divergence. The key idea is that the rollout prefix $\hat{o}_{<m}$ comes from the student itself, so the trajectory remains on-policy, while the teacher provides a stronger target by using the additional context r.

D-OPSD extends this idea to step-distilled text-to-image diffusion models (Jiang et al., 2026). In D-OPSD, the student follows its own few-step denoising trajectory conditioned on the text prompt, while the teacher is conditioned on richer multimodal context constructed from the text prompt and its paired real image. The student is then supervised along its own rollout states, preserving the original few-step inference behavior. Our goal is to instantiate this principle in few-step AR video generation. Compared with image diffusion, the on-policy state in AR video is more complex: it includes not only the current noisy latent and denoising timestep, but also the temporal KV cache produced by previously generated chunks. In the following section, we introduce OPSD-V, which uses real long videos as privileged temporal context to construct a cleaner teacher distribution, while keeping the student rollout and cache dynamics aligned with inference.

4 Method

4.1 Overview

We propose OPSD-V, a cache-aware on-policy self-distillation framework for post-training few-step causal AR video generators. Given a long training video divided into latent chunks $x_{1:N}^{\text{data}}$, we use the first chunk x_1^{data} as a shared real-video

prefix for both student and teacher. This prefix initializes the KV cache and anchors the scene, but does not participate in generation or loss computation. Starting from chunk $i = 2$, the student follows the original inference-time rollout: it denoises each chunk with the fixed few-step sampler, writes its own generated chunk into the KV cache, and continues from this self-generated temporal state.

As shown in Fig. 3, the key idea is to keep the student rollout fully on-policy while constructing a cleaner teacher distribution from real long-video context. For each generated chunk, the teacher is evaluated at the same student-visited noisy latents and denoising timesteps, but with a different temporal cache: older history is replaced by real-video chunks, while the most recent chunk remains student-generated to preserve autoregressive continuation. For example, at chunk $i = 7$, the student conditions on $x_1^{\text{data}}, \hat{x}_2^s, \hat{x}_3^s, \hat{x}_4^s, \hat{x}_5^s, \hat{x}_6^s$, whereas the teacher conditions on $x_1^{\text{data}}, x_2^{\text{data}}, x_3^{\text{data}}, x_4^{\text{data}}, x_5^{\text{data}}, \hat{x}_6^s$. This design provides dense denoising-level corrective targets on the student's own rollout states without changing the original few-step sampling path. We next describe the student on-policy rollout, the construction of the real-video teacher cache, and the dense distillation objective.

4.2 Student On-Policy Rollout

The student branch defines the on-policy rollout states used for training. We first initialize the student cache with the real first chunk x_1^{data}. Specifically, x_1^{data} is passed through the causal DiT to obtain its KV entries:

$$h_2^s = \mathrm{KV}_\theta(x_1^{\text{data}}), \tag{4}$$

where $\mathrm{KV}_\theta(\cdot)$ denotes the operation that converts a clean chunk into the corresponding cached key-value entries. This first chunk serves only as a shared context and is not generated by the student.

Starting from chunk $i = 2$, the student follows the exact inference-time procedure. Given the current student cache h_i^s, the student starts from Gaussian noise $z_{i,K}^s$ and denoises it with the fixed few-step sampler. At denoising step k, the student predicts a velocity field

$$\hat{v}_{i,k}^s = f_\theta(z_{i,k}^s, t_k, c, h_i^s), \tag{5}$$

where $z_{i,k}^s$ is the student noisy latent, t_k is the denoising timestep, and c is the text condition. The solver then updates the latent state by

$$z_{i,k-1}^s = \mathrm{sg}\left(\Phi(z_{i,k}^s, t_k, t_{k-1}, \hat{v}_{i,k}^s)\right), \qquad k = K, \ldots, 1. \tag{6}$$

Here $\mathrm{sg}(\cdot)$ denotes stop-gradient through the rollout transition, which prevents losses on later chunks from backpropagating through the entire autoregressive history. During rollout, we record all student states $\{z_{i,k}^s, \hat{v}_{i,k}^s\}_{k=1}^K$ for teacher evaluation and distillation.

After the final denoising step, the generated chunk is

$$\hat{x}_i^s = z_{i,0}^s. \tag{7}$$

It is then written into the student cache using the same cache update rule as inference:

$$h_{i+1}^s = h_i^s \oplus \mathrm{KV}_\theta(\hat{x}_i^s), \tag{8}$$

where $\oplus$ denotes appending the new KV entries to the existing cache.

Equivalently, before generating chunk i, the student cache contains the real first chunk followed by all previously generated student chunks:

$$h_i^s \equiv \mathrm{KV}_\theta(x_1^{\text{data}}) \oplus \mathrm{KV}_\theta(\hat{x}_2^s) \oplus \cdots \oplus \mathrm{KV}_\theta(\hat{x}_{i-1}^s). \tag{9}$$

Therefore, every supervised student prediction is conditioned on the same type of self-generated KV-cache state that the model will encounter during deployment.

4.3 AR-Consistent Real-Video Teacher Cache

The teacher branch provides cleaner corrective targets while preserving autoregressive continuation. Let $\bar{\theta}$ denote the teacher parameters. In our LoRA post-training setting, the base video generator is frozen, the student LoRA is trainable, and the teacher LoRA is maintained as an exponential-moving-average (EMA) copy of the student LoRA. The teacher is used only to produce stop-gradient targets.

For chunk i, the teacher is evaluated at the same student-visited noisy latent $z_{i,k}^s$ and timestep t_k, but with a different temporal cache h_i^t:

$$\hat{v}_{i,k}^t = f_{\bar{\theta}}(z_{i,k}^s, t_k, c, h_i^t). \tag{10}$$

Importantly, the teacher does not sample its own denoising trajectory. The denoising state remains on-policy because $z_{i,k}^s$ comes from the student rollout, while the teacher cache provides a cleaner temporal context for constructing the target velocity. In other words, the student trajectory determines where supervision is applied, and the teacher cache determines the corrective direction.

The teacher cache is constructed from real long-video chunks with an autoregressive consistency constraint. Before supervising chunk i, the student cache contains the real first chunk followed by previously generated student chunks:

$$h_i^s \equiv \mathrm{KV}_\theta(x_1^{\mathrm{data}}) \oplus \mathrm{KV}_\theta(\hat{x}_2^s) \oplus \cdots \oplus \mathrm{KV}_\theta(\hat{x}_{i-1}^s). \tag{11}$$

In contrast, the teacher cache replaces the older generated history with the corresponding real-video chunks, while keeping the most recent student-generated chunk:

$$h_i^t \equiv \mathrm{KV}_{\bar{\theta}}(x_1^{\mathrm{data}}) \oplus \mathrm{KV}_{\bar{\theta}}(x_2^{\mathrm{data}}) \oplus \cdots \oplus \mathrm{KV}_{\bar{\theta}}(x_{i-2}^{\mathrm{data}}) \oplus \mathrm{KV}_{\bar{\theta}}(\hat{x}_{i-1}^s). \tag{12}$$

For example, at chunk $i = 7$, the two caches are

$$\begin{aligned} h_7^s &: [x_1^{\mathrm{data}}, \hat{x}_2^s, \hat{x}_3^s, \hat{x}_4^s, \hat{x}_5^s, \hat{x}_6^s], \\ h_7^t &: [x_1^{\mathrm{data}}, x_2^{\mathrm{data}}, x_3^{\mathrm{data}}, x_4^{\mathrm{data}}, x_5^{\mathrm{data}}, \hat{x}_6^s]. \end{aligned} \tag{13}$$

This cache policy balances two goals. Replacing older history with real-video chunks reduces accumulated error contamination in the teacher context and provides a cleaner long-range temporal state. Keeping the most recent student-generated chunk prevents the teacher from becoming a fully teacher-forced oracle: the teacher still predicts how to continue from the student's latest generated state, which better matches autoregressive deployment. Both student and teacher use the same attention-sink cache mechanism with rolling relative RoPE. The sink positions are updated consistently as the rollout grows, so the difference between student and teacher predictions mainly comes from cache content rather than inconsistent positional indexing.

4.4 Dense Denoising-Level Objective

OPSD-V provides supervision at the fixed denoising steps used by the few-step sampler. For each supervised chunk-step pair (i, k), the student prediction $\hat{v}_{i,k}^s$ and the teacher target $\hat{v}_{i,k}^t$ are evaluated at the same student-visited noisy latent $z_{i,k}^s$ and timestep t_k, but under different temporal caches. We use a pure velocity matching objective:

$$\mathcal{L}_{\text{OPSD-V}} = \frac{1}{|\mathcal{S}|} \sum_{(i,k)\in\mathcal{S}} \left\| \hat{v}_{i,k}^s - \mathrm{sg}(\hat{v}_{i,k}^t) \right\|_2^2. \tag{14}$$

where $\mathrm{sg}(\cdot)$ stops gradients through the teacher prediction, and $\mathcal{S}$ denotes the set of supervised chunk-step pairs.

In our implementation, the underlying AR generator uses a four-step sampler, and we supervise all four denoising steps. We apply a rollout warm-up by excluding the first $M = 7$ chunks from the loss:

$$\mathcal{S} = \{(i, k) \mid i > M,\ k = 1, \ldots, K\}, \qquad M = 7,\ K = 4. \tag{15}$$

The warm-up chunks are still generated autoregressively and written into the student cache, but they do not contribute to the distillation loss. We set $M = 7$ because the Wan-based AR models used in our experiments are originally trained within a local window of seven chunks, corresponding to $3 \times 7 = 21$ latent frames. Starting the loss after this local horizon encourages supervision to focus on long-rollout states where accumulated cache degradation begins to appear. The loss is then applied to all subsequent chunks in the same contiguous long-video rollout.

4.5 Training Procedure

We summarize the training procedure of OPSD-V in Algorithm 1. For each long training video, both branches start from the same real first chunk. The student then performs autoregressive self-rollout to define the on-policy denoising states. After the rollout warm-up, the teacher cache is constructed for each supervised chunk with real-video older history and the most recent student-generated chunk, as defined in Eq. equation 12. The EMA teacher is evaluated at the same student-visited noisy latents, and the student is optimized with dense velocity matching on these supervised chunks.

Memory-efficient truncated backpropagation. A naive implementation would retain the computation graphs of all supervised chunk-step pairs until the complete long-video rollout is finished, causing activation memory to grow with both rollout length and the number of denoising steps. This is unnecessary in our formulation because the solver transition in Eq. equation 6 is stop-gradient and all KV-cache writes are performed without gradients. Consequently, the gradient of Eq. equation 14 decomposes over supervised pairs:

$$\nabla_\theta \mathcal{L}_{\text{OPSD-V}} = \sum_{(i,k)\in\mathcal{S}} \nabla_\theta \frac{\ell_{i,k}}{|\mathcal{S}|}, \qquad \ell_{i,k} = \left\| \hat{v}_{i,k}^s - \mathrm{sg}(\hat{v}_{i,k}^t) \right\|_2^2. \tag{16}$$

We therefore backpropagate each normalized term $\ell_{i,k}/|\mathcal{S}|$ immediately after its student forward pass and accumulate parameter gradients without updating the model. The current activation graph is then released before proceeding to the next denoising step. Teacher predictions for supervised chunks, warm-up student predictions, denoising transitions, and

Algorithm 1 Memory-Efficient Training Procedure of OPSD-V

Require: Long-video dataset $\mathcal{D}$, student model f_θ, EMA teacher $f_{\bar{\theta}}$, denoising steps K, warm-up chunks M

while not converged **do**
  Sample a contiguous long video and condition $(x^{\text{data}}_{1:N}, c) \sim \mathcal{D}$.
  Initialize the detached student cache from the shared prefix x^{data}_1.
  Initialize the accumulated parameter gradient: $g \leftarrow 0$.
  for $i = 2$ to N **do**
    Sample initial noise $z^s_{i,K} \sim \mathcal{N}(0, I)$.
    if $i > M$ **then**
      Construct h^t_i with real older history and the latest student-generated chunk, as in Eq. (12).
    end if
    for $k = K$ to 1 **do**
      if $i > M$ **then**
        Predict teacher target without gradients: $\hat{v}^t_{i,k} \leftarrow f_{\bar{\theta}}(z^s_{i,k}, t_k, c, h^t_i)$.
        Predict student velocity with gradients: $\hat{v}^s_{i,k} \leftarrow f_\theta(z^s_{i,k}, t_k, c, h^s_i)$.
        Compute $\ell_{i,k} \leftarrow \left\| \hat{v}^s_{i,k} - \text{sg}(\hat{v}^t_{i,k}) \right\|_2^2$.
        Backpropagate immediately: $g \leftarrow g + \nabla_\theta(\ell_{i,k}/|\mathcal{S}|)$.
        Release the current activation graph and detach all cache tensors.
      else
        Predict student velocity without gradients: $\hat{v}^s_{i,k} \leftarrow f_\theta(z^s_{i,k}, t_k, c, h^s_i)$.
      end if
      Update student latent without gradients: $z^s_{i,k-1} \leftarrow \text{sg}\Big(\Phi(z^s_{i,k}, t_k, t_{k-1}, \hat{v}^s_{i,k})\Big)$.
    end for
    Obtain generated chunk: $\hat{x}^s_i \leftarrow z^s_{i,0}$.
    Append $\hat{x}^s_i$ to the detached student cache.
  end for
  Clip g and update the student LoRA parameters θ once.
  Update the teacher LoRA parameters $\bar{\theta}$ with EMA.
end while

cache updates are all evaluated without gradients. The optimizer is stepped only once after the complete rollout, followed by the EMA teacher update. This online accumulation is mathematically equivalent to backpropagating the summed objective, while retaining at most one gradient-enabled student forward graph at a time. Thus, activation memory does not grow with the number of supervised chunk-step pairs; only the detached KV caches and accumulated parameter gradients persist across the rollout.

In our experiments, $K = 4$ and all denoising steps are supervised. We set $M = 7$, so the first seven chunks are still generated and written into the student cache, but they do not contribute to the loss. This warm-up lets the student enter long-rollout cache states before dense distillation begins. Although all four denoising steps receive supervision, their activation graphs are processed and released one at a time according to Eq. equation 16. At inference time, the teacher branch and real-video cache are removed; the model uses the original few-step AR sampling path.

5 Experiments

5.1 Experimental Setup

Training data. We construct a small customized long-video dataset for OPSD-V post-training. The dataset contains 3,800 videos, each approximately one minute in length and processed at 480p resolution. The videos cover diverse content, including natural landscapes, large camera motions, and human-centered scenes. We apply a lightweight filtering process based on optical flow and other basic quality cues to remove low-motion, unstable, or low-quality samples. All videos are encoded into latent chunks using the Wan2.1 VAE Wan Team (2025). Compared with short-clip training data, these long videos provide richer temporal context and are therefore suitable for constructing the real-video teacher cache in our framework. During training, we use the first chunk as the shared real-video prefix for both student and teacher, and perform autoregressive rollout on subsequent chunks.

Base models. We evaluate OPSD-V on two representative few-step AR video generators: Self-Forcing (Huang et al., 2025a) and LongLive (Yang et al., 2025b). Both models are built upon the Wan2.1-T2V-1.3B backbone Wan Team (2025). For Self-Forcing, we start from its official base checkpoint and train a LoRA adapter for our post-training. For LongLive,

Figure 4: Qualitative comparison on long autoregressive video generation. The first two examples are based on LongLive Yang et al. (2025b), and the last two examples are based on Self-Forcing Huang et al. (2025a). For each example, the first row shows the result generated by the original base model, and the second row shows the result after applying OPSD-V post-training. Each pair uses the same prompt, the same random seed, the same 4-step sampler, and the same attention-sink inference setting. Compared with the base models, OPSD-V better preserves motion dynamics and visual quality during long rollout, reducing common AR degradation artifacts such as weakened motion, blur, temporal flickering, and semantic drift in later chunks.

we use its official base model and released LoRA checkpoint, and continue post-training the LoRA adapter with OPSD-V. Evaluating on both models allows us to test whether OPSD-V can serve as a general post-training refinement method for different few-step AR video generators.

Post-training setting. The student branch uses the trainable LoRA adapter, while the teacher branch uses an exponential-moving-average (EMA) copy of the student LoRA with a decay rate of 0.9999. The teacher is only used

Table 1: Quantitative comparison on one-minute video generation. We generate 240 videos in total, including 120 prompts from MovieGenBench and 120 internal prompts from MeiBench, and evaluate all results with VBenchLong (Huang et al., 2025b). All methods are based on the same Wan2.1-T2V-1.3B backbone, use the same 4-step autoregressive inference path, and adopt the same attention-sink cache mechanism. Higher scores are better. OPSD-V improves Quality Score and Dynamic Degree on both LongLive and Self-Forcing without increasing inference cost.

Method	Params	NFE	Quality Score ↑	Dynamic Degree ↑	Semantic Score ↑
LongLive	1.3B	4	0.8138	0.5012	**0.4911**
LongLive + OPSD-V	1.3B	4	**0.8242**	**0.5890**	0.4904
Self-Forcing	1.3B	4	0.8259	0.6218	**0.4897**
Self-Forcing + OPSD-V	1.3B	4	**0.8389**	**0.6570**	0.4864

during training and is removed at inference time. We use the original four-step sampler of the base AR video models and supervise all four denoising steps with the velocity matching loss in Eq. equation 14. No additional DMD, reward, or adversarial loss is used. We train each model for 200 iterations on 24 H800 GPUs with a per-GPU batch size of 1. Training uses BF16 mixed precision, activation checkpointing, and fully sharded data parallelism. Together with the immediate per-step backward strategy described in Sec. 4.5, these choices enable long-rollout post-training without retaining the full rollout computation graph.

Rollout and cache construction. Each training sample is a contiguous long-video segment. For each rollout, we use 180 latent frames in total, divided into 60 chunks, with each chunk containing 3 latent frames. Both student and teacher are initialized from the same real first chunk. The student then follows the exact inference-time rollout, writing each generated chunk into its KV cache. The teacher cache is constructed with real-video chunks for older history and the most recent student-generated chunk for autoregressive continuation. We exclude the first $M = 7$ chunks from the distillation loss, corresponding to the original local training horizon of the Wan-based AR models. These warm-up chunks are still generated and written into the student cache, but only subsequent chunks contribute to the loss.

Evaluation protocol. We evaluate OPSD-V both qualitatively and quantitatively on one-minute video generation at 16 FPS. For qualitative comparison, we use the same prompt, random seed, sampler, and attention-sink inference setting for each pair of baseline and OPSD-V post-trained models. This allows us to directly compare the effect of our post-training objective under identical inference conditions. For quantitative evaluation, we use 240 prompts in total: the first 120 prompts from MovieGenBench (Polyak et al., 2024) and 120 internal prompts, which we refer to as MeiBench. Each method generates one video per prompt, and the resulting 240 videos are evaluated with VBenchLong (Huang et al., 2025b). We report averaged results over MovieGenBench and MeiBench. All compared models use the same 4-step AR inference path and the same inference-time KV-cache mechanism as their corresponding base models. Therefore, any improvement comes from post-training rather than additional inference computation.

5.2 Qualitative Comparison

Fig. 4 compares the original few-step AR generators with their OPSD-V post-trained counterparts under identical inference settings. Across both LongLive and Self-Forcing examples, the base models can produce plausible early chunks, but visual quality tends to degrade as autoregressive rollout continues. Typical failure modes include weakened motion, accumulated blur, local artifacts, and semantic or structural drift in later frames. After applying OPSD-V, the generated videos better preserve scene structure and motion dynamics over the full one-minute horizon. For LongLive, OPSD-V maintains sharper background details and more stable camera evolution in long landscape sequences. For Self-Forcing, OPSD-V produces more coherent object motion and reduces late-stage artifacts around the subject and background. These improvements are obtained without changing the sampler, number of denoising steps, or inference-time KV-cache mechanism, suggesting that cache-aware on-policy supervision improves the model's behavior on the states it naturally visits during deployment.

5.3 Quantitative Comparison

Tab. 1 reports the quantitative results averaged over MovieGenBench and MeiBench. Across both LongLive and Self-Forcing backbones, OPSD-V consistently improves Quality Score and Dynamic Degree while using the same model scale, 4-step inference path, and attention-sink cache mechanism. These results indicate that our cache-aware on-policy distillation improves long-video generation quality mainly by enhancing motion dynamics, rather than relying on extra inference computation. The Semantic Score remains comparable to the corresponding base models, with a slight decrease

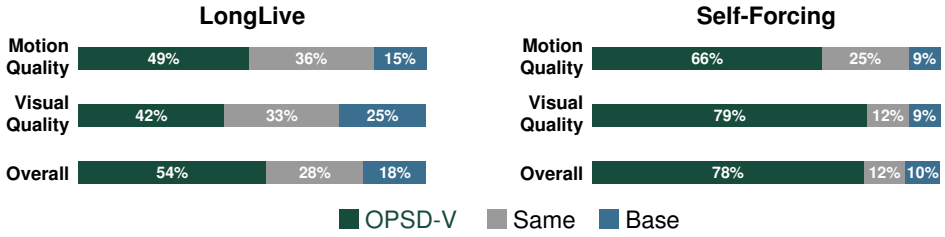

Figure 5: **User study preference results.** Ten participants compare 20 paired videos each, consisting of 10 LongLive pairs and 10 Self-Forcing pairs. Each pair contains a base-model video and its OPSD-V post-trained counterpart under the same prompt. For each pair, participants choose Model A, Model B, or Same for overall preference, motion quality, and visual quality. Each panel shows percentages over 100 judgments per criterion.

after post-training, suggesting a mild trade-off between more active motion generation and conservative semantic matching metrics.

We further conduct a user study with 10 participants. Each participant compares 20 paired videos, including 10 LongLive pairs and 10 Self-Forcing pairs. Each pair contains one video generated by the base model and one generated after applying OPSD-V, shown together with the corresponding prompt. For each pair, participants answer three questions: *Overall* preference considering all factors, *Motion Quality* focusing on smoothness, naturalness, and absence of jitter or discontinuity, and *Visual Quality* focusing on clarity, level of detail, and overall aesthetic quality. For each question, participants choose Model A, Model B, or Same when there is no perceptible difference. Fig. 5 reports the results separately for the two backbones. On LongLive, OPSD-V is favored over the base model in overall and motion-quality judgments, while visual quality shows a larger fraction of ties and base-model preferences. On Self-Forcing, OPSD-V receives strong preference across all three criteria. Aggregated over both backbones, OPSD-V is preferred in 66.0% of overall-quality comparisons, or 82.5% after excluding ties. Users also favor OPSD-V for motion quality (57.5%, 82.7% excluding ties) and visual quality (60.5%, 78.1% excluding ties). Overall, the quantitative results and human preferences support OPSD-V as an effective post-training refinement strategy for few-step AR video generation.

5.4 Ablation Study

We study two design choices that directly affect the stability of long-horizon post-training: the prediction space used for distillation and the trajectory on which the distillation targets are evaluated. Unless otherwise specified, all variants use the same training videos, four-step sampler, rollout length, cache construction, and LoRA post-training setup as our main method.

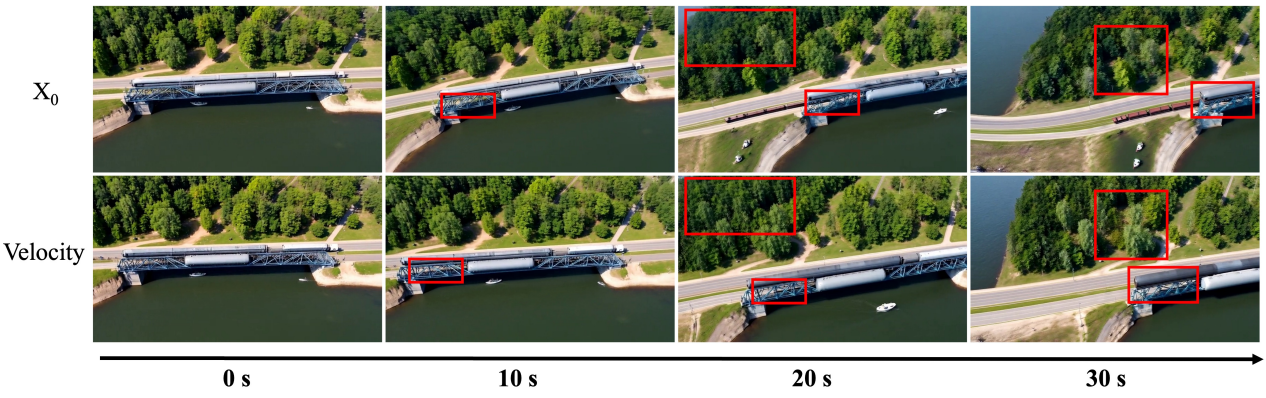

Figure 6: **Velocity versus clean-latent matching.** We compare distillation in the predicted clean-latent (x_0) space and in the native velocity space over a one-minute rollout. Both variants remain plausible at the beginning, but x_0 matching progressively smooths fine structures and introduces geometric degradation in the bridge truss and foliage. Velocity matching better preserves these details at later timestamps. Red boxes highlight representative regions.

Velocity versus clean-latent matching. Our default objective directly matches the student and teacher velocity predictions, as defined in Eq. equation 14. As an alternative, we convert both predictions into clean-latent estimates at every denoising step and optimize

$$\mathcal{L}_{x_0} = \frac{1}{|\mathcal{S}|} \sum_{(i,k)\in\mathcal{S}} \left\| \hat{x}^s_{0,i,k} - \mathrm{sg}(\hat{x}^t_{0,i,k}) \right\|_2^2 . \tag{17}$$

Under the flow parameterization, converting a velocity prediction to x_0 introduces a timestep-dependent scale. Consequently, an x_0-space MSE reweights prediction errors across the four fixed denoising steps and places relatively greater emphasis on high-noise states. As shown in Fig. 6, this variant produces increasingly smooth details during long rollout, particularly in the bridge structure and tree boundaries. Direct velocity matching instead supervises the model in its native prediction space and better preserves high-frequency structure. We therefore use velocity matching for all main experiments.

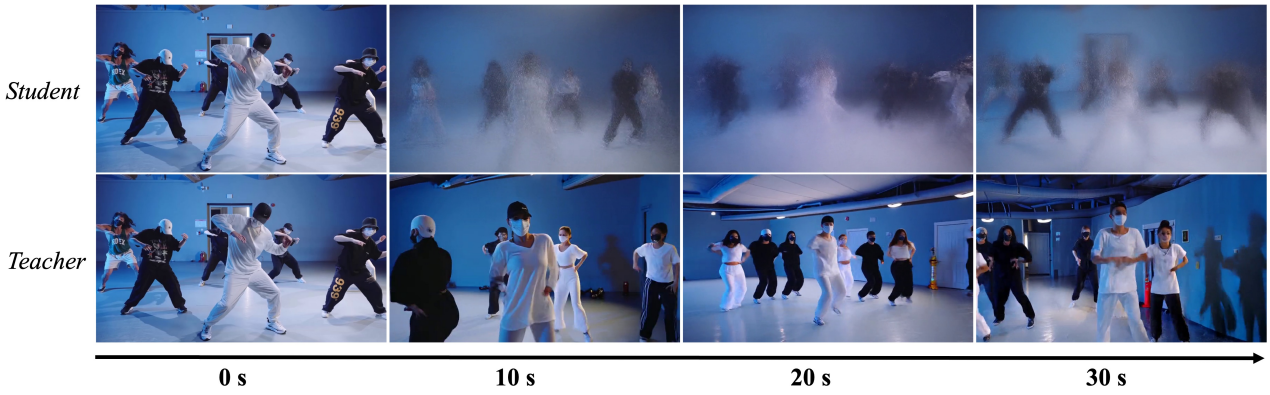

Figure 7: **Failure of teacher-trajectory supervision.** When distillation states are sampled from the teacher's own denoising trajectory, the teacher rollout remains sharp and coherent (bottom), but the student becomes severely blurred under its own autoregressive rollout (top). This discrepancy reveals the off-policy state mismatch introduced by teacher-trajectory supervision.

Student versus teacher trajectory. A central design choice in OPSD-V is that the teacher is evaluated at the noisy latents visited by the student, $z^s_{i,k}$, rather than sampling an independent denoising trajectory. We ablate this choice by constructing the distillation pairs from the teacher's own trajectory $z^t_{i,k}$. Although these teacher states yield visually cleaner targets, they are not the states that the student encounters when generating autoregressively at inference time. As a result, the student receives little supervision on how to recover from errors in its own denoising states, and the mismatch is repeatedly amplified after each generated chunk is written into the KV cache. Fig. 7 shows the resulting failure mode: the teacher trajectory remains clear, whereas the student rollout rapidly loses spatial detail and becomes dominated by blur. This result confirms that the teacher should provide the corrective prediction, but the student must determine the states on which that prediction is evaluated. We therefore keep the denoising trajectory fully on-policy in our final design.

6 Analysis and Future Work

Our work starts from a simple question: what should be used as the target distribution when post-training few-step autoregressive video generators? Existing DMD-style approaches usually rely on a short-clip diffusion model as the teacher distribution. While effective for accelerating generation, such a target is inherently limited by the temporal range and quality of the teacher model, making it difficult to provide reliable supervision for truly long autoregressive rollouts. A natural alternative is to directly exploit real long videos, which contain richer temporal structure and cleaner long-range dynamics.

However, incorporating real videos into few-step AR post-training is not straightforward. Direct teacher forcing would break the inference-time rollout distribution, while using real videos only as reconstruction targets would not correct the model on its own generated cache states. Our key observation is that on-policy self-distillation provides a suitable interface for this problem: real videos can be treated as privileged temporal context rather than direct output targets. In this way, the student still follows its own inference-time trajectory, while the teacher uses cleaner real-video cache states to provide dense denoising-level corrective supervision.

This perspective suggests a promising direction for future work. Since OPSD-V explicitly leverages real long-video data, its performance may further benefit from scaling the amount, diversity, and quality of training videos, as well as increasing training compute. Moreover, our current cache construction and loss design are only one possible instantiation of this idea.

More effective teacher-cache policies, adaptive supervision schedules, or stronger context-enhanced teachers may further improve long-horizon generation. We hope that OPSD-V can serve as a useful baseline and provide insight for future research on data-driven post-training of causal autoregressive video models.

7 Conclusion

We presented OPSD-V, a cache-aware on-policy self-distillation framework for post-training few-step autoregressive video generators. Instead of relying on short-clip teacher targets or fully teacher-forced training, OPSD-V supervises the student on its own inference-time rollout states while using real long-video context to construct cleaner teacher caches. This provides dense denoising-level correction under self-generated KV-cache states, improving long-horizon generation behavior without changing the original few-step inference path. Experiments on Self-Forcing and LongLive show that OPSD-V improves video quality and motion dynamics while preserving the same model scale and inference cost. We hope this work provides a useful step toward data-driven post-training for causal long-video generation.

References

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

Philip J. Ball, Jakob Bauer, Frank Belletti, Bethanie Brownfield, Ariel Ephrat, Shlomi Fruchter, Agrim Gupta, Kristian Holsheimer, Aleksander Holynski, Jiri Hron, Christos Kaplanis, Marjorie Limont, Matt McGill, Yanko Oliveira, Jack Parker-Holder, Frank Perbet, Guy Scully, Jeremy Shar, Stephen Spencer, Omer Tov, Ruben Villegas, Emma Wang, Jessica Yung, Cip Baetu, Jordi Berbel, David Bridson, Jake Bruce, Gavin Buttimore, Sarah Chakera, Bilva Chandra, Paul Collins, Alex Cullum, Bogdan Damoc, Vibha Dasagi, Maxime Gazeau, Charles Gbadamosi, Woohyun Han, Ed Hirst, Ashyana Kachra, Lucie Kerley, Kristian Kjems, Eva Knoepfel, Vika Koriakin, Jessica Lo, Cong Lu, Zeb Mehring, Alex Moufarek, Henna Nandwani, Valeria Oliveira, Fabio Pardo, Jane Park, Andrew Pierson, Ben Poole, Helen Ran, Tim Salimans, Manuel Sanchez, Igor Saprykin, Amy Shen, Sailesh Sidhwani, Duncan Smith, Joe Stanton, Hamish Tomlinson, Dimple Vijaykumar, Luyu Wang, Piers Wingfield, Nat Wong, Keyang Xu, Christopher Yew, Nick Young, Vadim Zubov, Douglas Eck, Dumitru Erhan, Koray Kavukcuoglu, Demis Hassabis, Zoubin Gharamani, Raia Hadsell, Aäron van den Oord, Inbar Mosseri, Adrian Bolton, Satinder Singh, and Tim Rocktäschel. Genie 3: A new frontier for world models. 2025.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. URL https://openai.com/research/video-generation-models-as-world-simulators.

Shengqu Cai, Weili Nie, Chao Liu, Julius Berner, Lvmin Zhang, Nanye Ma, Hansheng Chen, Maneesh Agrawala, Leonidas Guibas, Gordon Wetzstein, and Arash Vahdat. Mode seeking meets mean seeking for fast long video generation. In *ICML*, 2026.

Boyuan Chen, Diego Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *arXiv preprint arXiv:2407.01392*, 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, 2025. URL https://arxiv.org/abs/2504.13074.

Justin Cui, Jie Wu, Ming Li, Tao Yang, Xiaojie Li, Rui Wang, Andrew Bai, Yuanhao Ban, and Cho-Jui Hsieh. Self-forcing++: Towards minute-scale high-quality video generation. *arXiv preprint arXiv:2510.02283*, 2025.

Zhen Fang, Wenxuan Huang, Yu Zeng, Yiming Zhao, Shuang Chen, Kaituo Feng, Yunlong Lin, Lin Chen, Zehui Chen, Shaosheng Cao, and Feng Zhao. Flow-opd: On-policy distillation for flow matching models. *arXiv preprint arXiv:2605.08063*, 2026.

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.

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

Yinghui He, Simran Kaur, Adithya Bhaskar, Yongjin Yang, Jiarui Liu, Narutatsu Ri, Liam Fowl, Abhishek Panigrahi, Danqi Chen, and Sanjeev Arora. Self-distillation zero: Self-revision turns binary rewards into dense supervision. *arXiv preprint arXiv:2604.12002*, 2026.

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

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *arXiv preprint arXiv:2506.08009*, 2025a.

Ziqi Huang, Fan Zhang, Xiaojie Xu, Yinan He, Jiashuo Yu, Ziyue Dong, Qianli Ma, Nattapol Chanpaisit, Chenyang Si, Yuming Jiang, et al. Vbench++: Comprehensive and versatile benchmark suite for video generative models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026.

Dengyang Jiang, Xin Jin, Dongyang Liu, Zanyi Wang, Mingzhe Zheng, Ruoyi Du, Xiangpeng Yang, Qilong Wu, Zhen Li, Peng Gao, Harry Yang, and Steven Hoi. D-opsd: On-policy self-distillation for continuously tuning step-distilled diffusion models. *arXiv preprint arXiv:2605.05204*, 2026.

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

Weijie Kong, Qi Tian, Zijian Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Haodong Li, Shaoteng Liu, Zhe Lin, and Manmohan Chandraker. Rolling sink: Bridging limited-horizon training and open-ended testing in autoregressive video diffusion. *arXiv preprint arXiv:2602.07775*, 2026a.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026b.

Quanhao Li, Junqiu Yu, Kaixun Jiang, Yujie Wei, Zhen Xing, Pandeng Li, Ruihang Chu, Shiwei Zhang, Yu Liu, and Zuxuan Wu. Diffusionopd: A unified perspective of on-policy distillation in diffusion models. *arXiv preprint arXiv:2605.15055*, 2026c.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. *International Conference on Learning Representations*, 2023.

Dongyang Liu, Peng Gao, David Liu, Ruoyi Du, Zhen Li, Qilong Wu, Xin Jin, Sihan Cao, Shifeng Zhang, Hongsheng Li, and Steven Hoi. Decoupled dmd: Cfg augmentation as the spear, distribution matching as the shield. *arXiv preprint arXiv:2511.22677*, 2025a.

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*, 2025b.

Xiaofeng Mao, Shaohao Rui, Kaining Ying, Bo Zheng, Chuanhao Li, Mingmin Chi, and Kaipeng Zhang. Packforcing: Short video training suffices for long video sampling and long context inference. *arXiv preprint arXiv:2603.25730*, 2026.

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

William Peebles and Saining Xie. Scalable diffusion models with transformers. *arXiv preprint arXiv:2212.09748*, 2022.

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

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*, 2026.

Joonghyuk Shin, Zhengqi Li, Richard Zhang, Jun-Yan Zhu, Jaesik Park, Eli Shechtman, and Xun Huang. Motionstream: Real-time video generation with interactive motion controls. *arXiv preprint arXiv:2511.01266*, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Meituan LongCat Team, Xunliang Cai, Meng Cheng, Feng Gao, Zhe Kong, Jiamu Li, Le Li, Weiheng Li, Hongyu Liu, Shuai Tan, et al. Longcat-video-avatar 1.5 technical report. *arXiv preprint arXiv:2605.26486*, 2026a.

Robbyant Team, Zelin Gao, Qiuyu Wang, Yanhong Zeng, Jiapeng Zhu, Ka Leong Cheng, Yixuan Li, Hanlin Wang, Yinghao Xu, Shuailei Ma, Yihang Chen, Jie Liu, Yansong Cheng, Yao Yao, Jiayi Zhu, Yihao Meng, Kecheng Zheng, Qingyan Bai, Jingye Chen, Zehong Shen, Yue Yu, Xing Zhu, Yujun Shen, and Hao Ouyang. Advancing open-source world models. *arXiv preprint arXiv:2601.20540*, 2026b.

Wan Team. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025.

Lizhen Wang, Yongming Zhu, Zhipeng Ge, Youwei Zheng, Longhao Zhang, Tianshu Hu, Shiyang Qin, Mingshuang Luo, Jiaxu Zhang, Xin Chen, et al. Flowact-r1: Towards interactive humanoid video generation. *arXiv preprint arXiv:2601.10103*, 2026.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in neural information processing systems*, 36:8406–8441, 2023.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *arXiv*, 2023.

Shaoshu Yang, Zhe Kong, Feng Gao, Meng Cheng, Xiangyu Liu, Yong Zhang, Zhuoliang Kang, Wenhan Luo, Xunliang Cai, Ran He, and Xiaoming Wei. Infinitetalk: Audio-driven video generation for sparse-frame video dubbing, 2025a. URL https://arxiv.org/abs/2508.14033.

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. *arXiv preprint arXiv:2509.22622*, 2025b.

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. *arXiv preprint arXiv:2408.06072*, 2024.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, Ayaan Malik, Kyungmin Lee, William Liang, Nadun Ranawaka, Jiasheng Gu, Yinzhen Xu, Guanzhi Wang, Fengyuan Hu, Avnish Narayan, Johan Bjorck, Jing Wang, Gwanghyun Kim, Dantong Niu, Ruijie Zheng, Yuqi Xie, Jimmy Wu, Qi Wang, Ryan Julian, Danfei Xu, Yilun Du, Yevgen Chebotar, Scott Reed, Jan Kautz, Yuke Zhu, Linxi "Jim" Fan, and Joel Jang. World action models are zero-shot policies, 2026. URL https://arxiv.org/abs/2602.15922.

Hidir Yesiltepe, Tuna Meral, Adil Kaan Akan, Kaan Oktay, and Pinar Yanardag. Infinity-rope: Action-controllable infinite video generation emerges from autoregressive self-rollout. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 40256–40265, 2026.

Jung Yi, Wooseok Jang, Paul Hyunbin Cho, Jisu Nam, Heeji Yoon, and Seungryong Kim. Deep forcing: Training-free long video generation with deep sink and participative compression. *arXiv preprint arXiv:2512.05081*, 2025.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T. Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. *arXiv preprint arXiv:2311.18828*, 2023.

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

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

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

Jingran Zhang, Ning Li, Yuanhao Ban, Andrew Bai, and Justin Cui. Reward-forcing: Autoregressive video generation with reward feedback. *arXiv preprint arXiv:2601.16933*, 2026.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*, 2026.

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-sora: Democratizing efficient video production for all. *arXiv preprint arXiv:2412.20404*, 2024.

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. *arXiv preprint arXiv:2602.02214*, 2026.