# 4DStreamCtrl: Interactive Video Generation with Online 4D Control

**Anonymous authors**
Paper under double-blind review

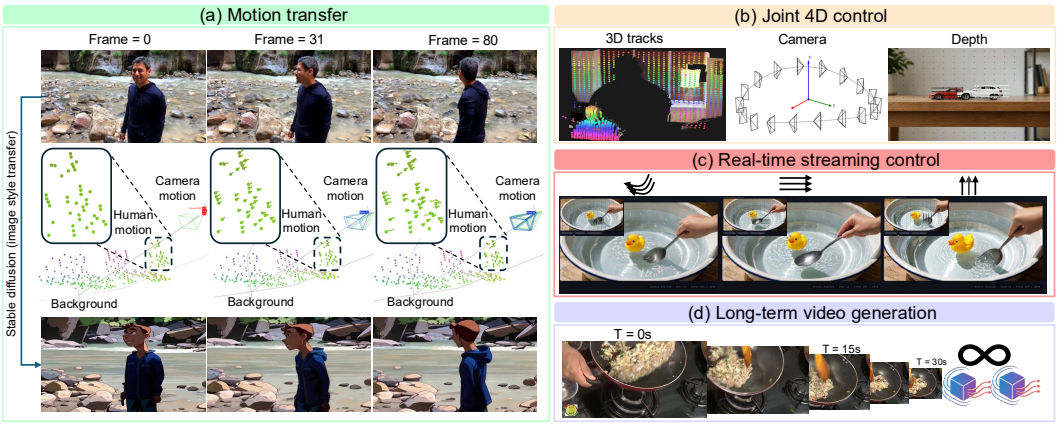



Figure 1: **Overview of 4DStreamCtrl capabilities. (a) Motion transfer**. Given a source video (top row), we decompose it into camera motion, human motion, and background structure (middle row), which transfer to a new scene via image style transfer while preserving the original 3D-consistent motion (bottom row). **(b) Joint 4D control**. 3D point tracks, camera parameters, and depth are unified into a single representation, enabling simultaneous control of camera and object motion. **(c) Real-time streaming control**. Users steer the generation with control signals (*e.g.*, dragging objects) as the video streams. **(d) Long-term video generation**. Through causal autoregressive distillation, 4DStreamCtrl generates arbitrarily long, temporally coherent videos at a memory cost independent of length.

## Abstract

Generative video models now synthesize footage nearly indistinguishable from reality. Their promise as interactive tools hinges on fine-grained control of how objects and the camera move over time, yet each existing approach captures only part of this: camera-parameter methods steer the viewpoint but cannot move objects, 2D-trajectory methods act in the image plane and ignore depth and occlusion, and recent 3D methods add geometry but run only offline at a fixed length. In particular, none combines 3D-consistent control of both camera and objects with real-time, streaming generation. Here we show that camera motion, object trajectories, and depth can be unified into a single 3D point-track representation, from which one model performs joint camera and object control, depth editing, and motion transfer in a single forward pass. To learn this interface at scale, we mine in-the-wild video for 3D motion supervision, yielding `OpenVidHD-Motion3D`, and encode it with a lightweight *Geometric Motion Head* that plugs into a pretrained video diffusion model. Because this encoder is temporally separable, we distill the model into a causal streaming student that generates arbitrarily long video in four denoising steps at memory independent of length. This unified design surpasses prior camera-only, 2D, and offline-3D methods in motion-control precision while covering modalities they address only in isolation. 4DStreamCtrl runs at 20 FPS on a single high-end GPU for 480p video and stays temporally coherent over hundreds of frames, enabling, to our knowledge, interactive 4D-controllable streaming generation for the first time. More broadly, grounding generation in explicit 3D geometry with efficient causal inference points toward interactive world models with closed-loop spatiotemporal control, from controllable simulators to real-time visual imagination for embodied agents.



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

# 1 INTRODUCTION

The ultimate goal of controllable video generation is to give users precise command over how objects move, how the camera behaves, and how these dynamics evolve, all in real time. Diffusion-based video models have advanced rapidly (Ho et al., 2022; Peebles & Xie, 2023), and open-weight systems such as Wan (Wan et al., 2025) and HunyuanVideo (Kong et al., 2024) now produce footage of striking realism. Yet their dominant interface, natural language with an optional reference image, underspecifies motion: "the camera orbits the object while the cup slides left" admits many unintended renderings, leaving practitioners in "render-and-wait" loops (Kong et al., 2024; Wan et al., 2025; Gu et al., 2025; Lee et al., 2026). Work that injects explicit motion cues splits across two fronts that no method has united. The first is *what the control signal can express*: camera-parameter methods (He et al., 2024; Wang et al., 2024b) steer the viewpoint but cannot move objects; 2D-trajectory methods (Wu et al., 2024b; Geng et al., 2025; Teng et al., 2023; Li et al., 2025) add object drag but, confined to the image plane, cannot encode depth or occlusion and break under parallax; only recent 3D-aware methods (Gu et al., 2025; Wang et al., 2025) restore full geometry via 3D point tracks. The second is *how generation runs*: most of them, 3D-aware included (Gu et al., 2025; Lee et al., 2026), are offline and fixed-length, producing a clip in one slow pass with no way to respond mid-generation. Missing is a method complete on the first front and interactive on the second: 3D-consistent control of both camera and objects, delivered as a real-time, steerable stream.

These shortfalls share one cause: no motion representation is at once geometrically complete and incrementally constructible. We present 4DStreamCtrl (Fig. 1), built on the insight that *3D point tracks and camera geometry can be unified into a single conditioning interface* expressing object trajectories, camera motion, and depth together. Projecting camera extrinsics onto background point tracks and encoding per-point depth alongside spatial coordinates makes this interface subsume camera, object, and depth control as special cases of one 3D-track signal, resolving both fronts. For expressiveness, a signal that jointly carries geometry, per-track identity, and depth dispels the ambiguity that defeats image-plane methods, letting one model, in a single forward pass, perform joint camera and object manipulation, cross-subject motion transfer, and depth-aware editing. For interactivity, the same representation admits lossless segment-by-segment encoding, the property that makes streaming possible.

To learn this interface at scale, we build `OpenVidHD-Motion3D`, the first large-scale 3D motion dataset for in-the-wild video, by mining OpenVid-1M (Nan et al., 2025) and annotating it with SpatialTrackerV2 (Xiao et al., 2025); after filtering it retains roughly 0.4M clips, each with video up to 1080p, a $32 \times 32$ grid of 3D point tracks, and per-frame camera intrinsics and extrinsics. A lightweight *Geometric Motion Head* encodes each trajectory's coordinates $(x, y, z)$, a sinusoidal point-identity embedding, and monocular depth into dense features on the VAE latent grid, fused with noisy latents by channel concatenation before the Wan2.2 TI2V-5B DiT backbone (Wan et al., 2025). Because its scatter-add rasterization and strided Conv3D run independently per chunk, encoding one segment is bit-identical to encoding the full sequence, so the offline teacher and streaming model share one motion pathway with no architectural change.

Sharing the encoder is necessary but not sufficient: the teacher's 50-step bidirectional denoising sees the whole clip at once, making it slow and unable to honor signals that arrive mid-playback. We therefore distill it into a causal autoregressive student following the self-forcing (Huang et al., 2025) and DMD (Yin et al., 2024b;a) paradigm adapted to our 3D-track interface. The student replaces bidirectional attention with a block-wise causal variant (Chen et al., 2024), keeps a fixed *attention sink* (Xiao et al., 2024) on the first-frame latents for global coherence and a *local sliding window* over recent chunks, and denoises each chunk in 4 steps, a $12.5\times$ reduction. With KV-cache reuse and memory independent of video length, it streams arbitrarily long 480p video at 20 FPS on a single high-end GPU, letting a user edit 3D trajectories on the fly while frames stream back and closing the loop between human intent and generation for the first time in a 4D-controllable setting.

Extensive experiments show that 4DStreamCtrl attains state-of-the-art motion-control precision and 3D consistency, outperforming prior trajectory- and camera-conditioned methods (Wang et al., 2024b; Shin et al., 2026; Burgert et al., 2025; Gu et al., 2025; Wang et al., 2025; Li et al., 2025), while its streaming student sustains coherent 350-frame clips (14.6 s at 24 FPS) at memory independent of length. Our contributions are threefold:

- A **unified 3D-track conditioning interface** that expresses camera motion, object trajectories, and depth as one signal, enabling joint control, cross-subject motion transfer, and depth-aware editing in a single forward pass.
- **`OpenVidHD-Motion3D`**, the first large-scale in-the-wild 3D motion dataset (roughly 0.4M clips with 3D tracks and camera parameters), and a lightweight, temporally separable *Geometric Motion Head* that injects these signals into a pretrained diffusion backbone.
- A **causal streaming distillation** that turns the 50-step teacher into a 4-step, constant-memory student, yielding, to our knowledge, the first real-time 4D-controllable interactive video generation on a single GPU.

## 2 Related Work

**Motion-conditioned video generation.** Video diffusion offers a scalable paradigm for generative video modeling (Ho et al., 2022; Peebles & Xie, 2023), and recent systems inject explicit motion cues to overcome the underspecification of language-only conditioning. Representative directions include object motion control (Wang et al., 2024b), entity-centric drag interfaces (Wu et al., 2024b), point-based non-rigid editing (Teng et al., 2023), and trajectory prompts that steer diffusion with sparse or dense point tracks (Geng et al., 2025; Wang et al., 2025; Li et al., 2025); open-weight backbones such as Wan (Wan et al., 2025), including the Wan2.2 TI2V-5B text-and-image-to-video checkpoint, make such interfaces practical to fine-tune. Most of these condition on *image-plane* motion, however, and recent 3D-aware generators (Gu et al., 2025) that instead use 3D point tracks remain restricted to offline, fixed-length synthesis. We study 3D trajectory conditioning with **joint object and camera control** to resolve the ambiguity between object and viewpoint motion, and target **interactive streaming** rather than one-shot clip generation.

**Scalable 3D supervision from video.** Recovering consistent geometry and cameras from monocular or multi-view video is long-standing, with classical structure-from-motion (Schönberger & Frahm, 2016) and learning-based correspondence and reconstruction models (Wang et al., 2024a) improving robustness in the wild, while modern long-range trackers supply scalable point-dynamics signals (Karaev et al., 2025) and large text–video datasets enable training at million-pair scale (Nan et al., 2025). We connect these ingredients by constructing **in-the-wild 3D-track supervision with camera intrinsics and extrinsics**, turning internet video into structured geometric motion signals for training trajectory-conditioned generators.

**Few-step distillation and streaming video generation.** Iterative diffusion sampling is too costly for closed-loop interaction, motivating few-step distillation: progressive distillation (Salimans & Ho, 2022) and distribution matching (Yin et al., 2024b;a) compress a bidirectional teacher into a handful of denoising steps. In parallel, autoregressive and chunk-based generators (Yu et al., 2023; Kondratyuk et al., 2024; Yin et al., 2025; Huang et al., 2025) synthesize frame blocks sequentially over a rolling context to enable streaming rather than one-shot generation, with a retained *attention sink* (Xiao et al., 2024) stabilizing long rollouts. Closest to our setting, MotionStream (Shin et al., 2026) distills a streaming controllable generator, but conditions on 2D image-plane tracks and so inherits their depth and occlusion ambiguity. We instead distill a causal student over our **3D-track interface**, carrying 3D-consistent joint control of camera and objects into real-time streaming and moving closer to interactive world-model deployment (Bruce et al., 2024; Yang et al., 2024) than one-shot full-clip sampling.

## 3 Method

We aim to learn a video generator for 4D control that (i) is trained on in-the-wild 3D motion supervision mined from large-scale internet video, (ii) conditions a pretrained text-and-image-to-video (TI2V) diffusion backbone on 3D motion features (including depth), and (iii) is distilled into a causal streaming student for interactive control at high frame rates.

### 3.1 3D Track Dataset Construction

**Source videos.** We start from OpenVid-1M (Nan et al., 2025), a large collection of text–video pairs with diverse in-the-wild content. For each clip, we run a fixed offline pipeline to extract per-frame

3D point trajectories together with camera intrinsics and extrinsics, yielding scalable supervision without manual motion annotation.

**3D tracking.** We extract dense 3D tracks with SpatialTrackerV2 (Xiao et al., 2025), a feed-forward monocular 3D point tracker that jointly estimates scene geometry, camera motion, and point dynamics. For each clip, we sample query points on a $32 \times 32$ grid in the first frame and propagate them across all subsequent frames, yielding per-point trajectories in a consistent world coordinate system together with per-frame camera intrinsics $\mathbf{K}_t$ and world-to-camera extrinsics $\mathbf{T}_t$. We then transform each world-space point into the corresponding camera frame via $\mathbf{T}_t$ and store the resulting per-point 3D trajectories $\mathbf{x}_{n,t} \in \mathbb{R}^3$ in camera space, where $n$ indexes tracks and $t$ indexes frames.

Internet videos contain cuts, extreme motion blur, and tracker failures. We discard unreliable and short sequences and keep only high-definition pairs, retaining approximately 0.4M clips after filtering. We normalize camera intrinsics to a canonical resolution and normalize depth to $[0, 1]$ for stable training. Each training example therefore comprises RGB frames, a text prompt, 3D tracks, and camera parameters. We refer to this collection as `OpenVidHD-Motion3D`; it provides structured conditioning targets for training and can also supply control signals at inference time. Full details of the data pipeline and teacher training configuration are given in Section B.1.

### 3.2 3D Motion-Conditioned Video Model

**Overview.** Our generator is based on a TI2V diffusion transformer (Wan2.2 TI2V-5B) (Wan et al., 2025); Fig. 2 provides an overview. Inspired by MotionStream (Shin et al., 2026), we inject motion by building a compact motion feature tensor on the VAE latent grid and fusing it with noisy latents *before* DiT patch embedding via **channel concatenation**. For our 4D control task, we extend this design in two ways: (i) 3D tracks are projected to normalized image coordinates for rasterization; and (ii) scalar depth is encoded in a **separate branch** rather than fused into per-track identity features prior to scattering, so geometric cues stay disentangled from track identity.

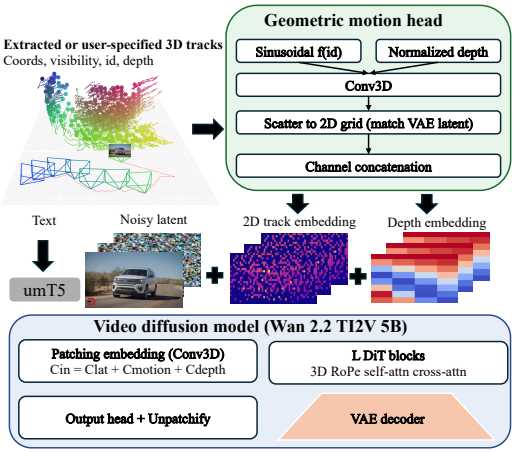



Figure 2: **Overview of 4DStreamCtrl.** Geometric 3D tracks and text are encoded into 2D and depth motion features, fused with noisy latents in a DiT with 3D RoPE, then decoded to video for precise trajectory, camera, and motion control.

**Notation.** For each batch sample $b \in \{1, \ldots, B\}$, we are given $N$ point tracks in camera space (Section 3.1). From each camera-space point $\mathbf{x}_{n,t} = (x, y, z)^\top$, we obtain normalized image coordinates and depth via

$$\mathbf{p}_{n,t} = \pi(\mathbf{K}_t, \mathbf{x}_{n,t}) \in [0, 1]^2, \qquad d_{n,t} = z, \tag{1}$$

where $\pi$ denotes perspective projection followed by resolution normalization. Track $n$ has an integer id $a_n$ and, at each frame $t \in \{1, \ldots, T\}$, provides image coordinates $(p^x_{n,t}, p^y_{n,t})^\top$ and an optional scalar depth $d_{n,t}$. Let $(H_\ell, W_\ell)$ denote the latent spatial resolution aligned with the VAE tokenizer grid, and let $F$ denote the number of latent frames fed to the DiT. The motion encoder operates on $T$ raw frames and temporally compresses features to length $F$: the first frame maps to a dedicated latent frame while every subsequent group of four frames is compressed into one, giving $F = (T-1)/4+1$.

**Per-track sinusoidal embeddings.** Each track id $a_n$ is mapped to a fixed sinusoidal positional encoding $\boldsymbol{\phi}_n = \mathrm{PE}(a_n) \in \mathbb{R}^D$, using the standard 1D construction (even $D$, log-spaced frequencies) rather than a learned embedding table. The resulting $\boldsymbol{\phi}_n$ is shared across all time steps and depends only on track identity.

**Rasterization on the latent grid.** We map normalized coordinates to discrete latent indices by nearest-neighbor quantization,

$$x_{n,t} = \left\lfloor p^x_{n,t} \cdot \max(W_\ell - 1,\, 1) \right\rceil, \qquad y_{n,t} = \left\lfloor p^y_{n,t} \cdot \max(H_\ell - 1,\, 1) \right\rceil, \tag{2}$$

and clamp indices to $[0, W_\ell - 1]$ and $[0, H_\ell - 1]$. We initialize a track raster $\mathbf{C} \in \mathbb{R}^{B\times D\times T\times H_\ell\times W_\ell}$ to zero and scatter-add features at occupied cells:

$$\mathbf{C}_{b,d,t,y,x} \;=\; \sum_{n=1}^{N} \boldsymbol{\phi}_{n,d}\, \mathbb{I}[x = x_{n,t},\, y = y_{n,t}]\,, \tag{3}$$

where $\mathbb{I}[\cdot]$ is the indicator function and multiple tracks landing in the same cell $(t, y, x)$ accumulate by summation.

**Track branch.** The raster $\mathbf{C}$ is processed by a lightweight shared Conv3D head,

$$\mathbf{F}^{\text{trk}} = \mathcal{H}_{\text{trk}}(\mathbf{C}) = \text{Conv3D}_{1\times1\times1}\Big(\text{SiLU}\big(\text{Conv3D}_{4\times1\times1}(\mathbf{C})\big)\Big), \tag{4}$$

where $\text{Conv3D}_{4\times1\times1}$ uses stride $(4, 1, 1)$ for $\times4$ temporal compression along the frame axis and the final $1 \times 1 \times 1$ layer maps to $C_{\text{trk}}$ channels.

**Depth branch.** When enabled, each space–time sample may provide depth $d_{n,t}$. We map depth to disparity $\tilde{d}_{n,t} = 1/(d_{n,t} + \varepsilon)$ for numerical stability, embed it via a linear map $\mathbf{e}^{\text{dep}}_{n,t} = \mathbf{W}_d\, \tilde{d}_{n,t} + \mathbf{b}_d \in \mathbb{R}^{D_d}$, and rasterize onto the same indices $(t, y_{n,t}, x_{n,t})$ as in Eq. (3) to obtain $\mathbf{C}^{\text{dep}} \in \mathbb{R}^{B\times D_d\times T\times H_\ell\times W_\ell}$. A separate Conv3D head $\mathcal{H}_{\text{dep}}$ with the same $(4, 1, 1)$ temporal structure produces $\mathbf{F}^{\text{dep}} \in \mathbb{R}^{B\times C_{\text{dep}}\times F\times H_\ell\times W_\ell}$. When depth is unavailable at runtime, we set $\mathbf{F}^{\text{dep}} = \mathbf{0}$, keeping the DiT input channel width fixed.

**Fusion with noisy latents.** Let $\mathbf{z} \in \mathbb{R}^{B\times C_{\text{lat}}\times F\times H_\ell\times W_\ell}$ denote noisy VAE latents. We form the motion conditioning tensor $\mathbf{F}^{\text{cond}} = \mathbf{F}^{\text{trk}} \,\|\, \mathbf{F}^{\text{dep}}$ by channel concatenation, then concatenate it with $\mathbf{z}$ to obtain $\tilde{\mathbf{z}} = \mathbf{z} \,\|\, \mathbf{F}^{\text{cond}} \in \mathbb{R}^{B\times(C_{\text{lat}}+C_{\text{trk}}+C_{\text{dep}})\times F\times H_\ell\times W_\ell}$, which is fed into the DiT backbone.

**Training objective.** Let $\mathbf{x}_0 \in \mathbb{R}^{B\times C_{\text{lat}}\times F\times H_\ell\times W_\ell}$ denote clean VAE latents of a training clip. We fine-tune the pretrained TI2V diffusion transformer with the same flow-matching supervision used by Wan (Wan et al., 2025). For each step we sample a flow time $\tau$ (equivalently a noise level $\sigma_\tau$), draw Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and form the interpolated latent $\mathbf{x}_\tau = (1 - \sigma_\tau)\mathbf{x}_0 + \sigma_\tau\boldsymbol{\epsilon}$. The network predicts a velocity field $\mathbf{v}_\theta(\mathbf{x}_\tau, \tau, \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{img}}, \mathbf{F}^{\text{cond}})$, where $\mathbf{c}_{\text{text}}$ is the text embedding, $\mathbf{c}_{\text{img}}$ encodes the first frame for TI2V, and $\mathbf{F}^{\text{cond}}$ is the motion tensor built from 3D tracks (and depth when available). Training minimizes a weighted mean-squared error to the flow target $\mathbf{v} = \boldsymbol{\epsilon} - \mathbf{x}_0$,

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{\tau,\boldsymbol{\epsilon}}\Big[w(\tau)\,\big\|\mathbf{v}_\theta(\mathbf{x}_\tau, \tau, \mathbf{c}_{\text{text}}, \mathbf{c}_{\text{img}}, \mathbf{F}^{\text{cond}}) - \mathbf{v}\big\|_2^2\Big]\,, \tag{5}$$

with $w(\tau)$ following the scheduler's timestep weighting. For TI2V, the first-frame latent is held fixed when constructing $\mathbf{x}_\tau$, and the loss is evaluated only on future latent frames.

**Parameter-efficient fine-tuning.** To preserve the pretrained generative prior, we adopt LoRA (Hu et al., 2022) on the DiT while keeping its base weights frozen. Low-rank adapters are inserted into the attention and feed-forward projections (i.e., $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_o$ and FFN layers). In parallel, the motion pathway, comprising the track and depth rasterizers and Conv3D heads ($\mathcal{H}_{\text{trk}}$, $\mathcal{H}_{\text{dep}}$), the depth projector $\mathbf{W}_d$, and the widened patch-embedding input, is trained end-to-end.

**Two-stage training schedule.** To stabilize optimization we use a coarse-to-fine curriculum: Stage A trains on short, low-resolution clips (256p, 49 frames) to learn coarse alignment between 3D motion cues and video dynamics at low compute, and Stage B continues at higher resolution and longer duration (480p, 81 frames) to refine motion–appearance alignment and long-horizon consistency. Ablations in Section 4.4 confirm its effectiveness.

### 3.3 Streaming Distillation for Interactive 4D Control

The bidirectional teacher (Section 3.2) produces high-quality 3D-aligned video but requires around 50 denoising steps over the full clip, far too slow for interaction. We distill it into a causal 4-step student following the self-forcing paradigm (Huang et al., 2025; Shin et al., 2026). The student reuses the teacher's motion encoder and channel-fusion pathway but replaces bidirectional self-attention with **block-wise causal attention** (Chen et al., 2024): latents are partitioned into chunks $\{\mathbf{z}^i\}_{i=1}^{L}$, and each chunk attends only to its own tokens, an attention sink anchored on the first frame (see Section C.4 for its effect), and a local window of preceding chunks, with a rolling KV cache ensuring

constant compute and memory. A key enabler is that our motion encoder is **temporally separable by construction**: the per-frame scatter-add rasterization (Eq. (3)) and the non-overlapping $(4, 1, 1)$ Conv3D head make segment-by-segment encoding bit-identical to full-sequence encoding, requiring no architectural change for streaming.

Following CausVid (Yin et al., 2025), we first initialize the student on 4 k noise–clean ODE pairs from the teacher to stabilize few-step denoising, then train with the DMD objective (Yin et al., 2024b) under temporal self-rollout: each chunk is denoised conditioned on the student's *own* prior outputs via KV cache, mirroring inference. Let $\mathbf{x}$ be the generator's clean rollout output and $\mathbf{x}_\tau = (1-\sigma_\tau)\mathbf{x} + \sigma_\tau \boldsymbol{\epsilon}$ its noised version. A frozen real score and a learnable fake score are jointly optimized,

$$\mathcal{L}_{\text{critic}} = \left\| \hat{\mathbf{x}}_0^{\text{fake}}(\mathbf{x}_\tau, \tau, \mathbf{c}) - \mathbf{x} \right\|^2, \qquad \nabla_{\mathbf{x}} \mathcal{L}_{\text{gen}} = \hat{\mathbf{x}}_0^{\text{fake}}(\mathbf{x}_\tau, \tau, \mathbf{c}) - \hat{\mathbf{x}}_0^{\text{real}}(\mathbf{x}_\tau, \tau, \mathbf{c}), \tag{6}$$

driving the student's output distribution toward the teacher's. At inference, generation proceeds chunk by chunk, and after each decoded chunk the motion encoder recomputes $\mathbf{F}^{\text{cond}}$ from the latest 3D tracks and depth, so users can revise control at any time. Further details are in Section B.2.

## 4 Experiments

We evaluate the 3D track-conditioned teacher and its distilled streaming student on three regimes that reflect the paper's core claims: (i) joint object and camera control, (ii) motion transfer, and (iii) interactive streaming control. Fig. 3 gives an overview of the qualitative results.

### 4.1 Joint Object and Camera Control

**Setup.** We evaluate on 30 in-the-wild videos from the DAVIS validation set (Perazzi et al., 2016), selected for challenging coupled camera and object motion. Each method receives the same first frame and text prompt where applicable, with conditioning in its native interface (*e.g.*, 2D tracks, 2D flow, or 3D tracks); our models take text, image, and 3D tracks, and we report both the full teacher and the causal streaming variant. We compare against recent motion-conditioned generators spanning 2D-track, 2D-flow, and 3D-track conditioning: Image Conductor (Li et al., 2025), Go-With-The-Flow (Burgert et al., 2025), Diffusion-As-Shader (Gu et al., 2025), ATI (Wang et al., 2025), and MotionStream (Shin et al., 2026), of which only Diffusion-As-Shader also consumes 3D tracks. Backbones and input modalities are listed in Table 1.

We report reconstruction fidelity against ground-truth DAVIS frames via PSNR and SSIM (higher is better), perceptual quality via LPIPS (Zhang et al., 2018) (lower is better), motion faithfulness via end-point error (EPE) between predicted and target motion (lower is better), and inference throughput (FPS) at each method's default resolution and step count.

**Results.** Our teacher attains the best motion alignment on DAVIS, with the lowest EPE (5.29) and LPIPS (0.404) and the highest SSIM (0.479). The gain from geometry is clearest under a matched backbone: on Wan 2.2-5B, MotionStream reaches EPE 7.86 and LPIPS 0.427, so replacing 2D with 3D tracks cuts EPE by roughly 33% and LPIPS by 5%. Weaker image-plane interfaces fall much further behind (ATI 17.41, Go-With-The-Flow 41.99, Image Conductor 91.64), while MotionStream on the smaller Wan 2.1-1.3B backbone is our closest competitor on EPE (5.35) and SSIM (0.477) yet still trails on perceptual quality. Diffusion-As-Shader also consumes 3D tracks but runs at 0.29 FPS and remains far behind on EPE (40.23) and LPIPS (0.483), indicating that 3D conditioning alone is insufficient without our geometric motion head and training recipe.

PSNR is the one metric on which we do not lead (16.04 against 16.61): pixel-wise PSNR rewards exact reproduction and so penalizes the plausible variation a controllable generator introduces where the tracks leave content underspecified, such as disoccluded surfaces and objects entering the frame. We therefore treat EPE and LPIPS as the primary indicators of control quality and report PSNR for completeness.

Our causal student reaches 20.6 FPS, ahead of MotionStream Causal on both Wan 2.1-1.3B (16.7) and Wan 2.2-5B (10.4), while its EPE (5.48) stays within 4% of our own teacher and below every causal baseline. Distillation therefore preserves 3D-track control rather than trading it for speed: the teacher sets the best SSIM, LPIPS, and EPE trade-off, and the student gives the strongest combination of throughput and track accuracy among Wan 2.2 systems.




Figure 3: **Qualitative results of 4DStreamCtrl across diverse control modes.** Each group demonstrates one control capability: **Object control** steers the trajectory of individual objects; **Camera control** adjusts viewpoint and camera movement; **Object and camera joint control** coordinates object dynamics with camera motion; **Depth control** conditions generation on per-point depth to resolve occlusion and depth ordering; and **Motion transfer** carries motion from a reference video onto new content. Frames run in temporal order from left to right. Point trajectories are overlaid for visualization; in the depth control group, white points mark trajectories at shallower depths and black points those at greater depths.

### 4.2 Motion Transfer

Given a source video, we extract its 3D tracks and camera parameters with SpatialTrackerV2, restyle the first frame into a new target appearance with Stable Diffusion XL, and condition our model on the stylized frame together with the extracted motion. As shown in the bottom group of Fig. 3, the generated video reproduces the original 3D-consistent motion, including human body dynamics, camera motion, and background parallax, while adopting an entirely new visual style and scene content. The unified 3D-track representation thus disentangles motion from appearance: the geometric

Table 1: **Results on joint object and camera control.** We evaluate motion-conditioned methods on 30 in-the-wild videos from the DAVIS validation set that contain challenging camera and object movement. Best results are in **bold**, second best are underlined.

| Method | Backbone | Input | FPS | DAVIS Validation Set | | | |
|---|---|---|---|---|---|---|---|
| | | | | PSNR↑ | SSIM↑ | LPIPS↓ | EPE↓ |
| Image Conductor (Li et al., 2025) | AnimateDiff | T, I, 2D Track | 2.98 | 11.30 | 0.214 | 0.664 | 91.64 |
| Go-With-The-Flow (Burgert et al., 2025) | CogVideoX-5B | T, I, 2D Flow | 0.60 | 15.62 | 0.392 | 0.490 | 41.99 |
| Diffusion-As-Shader (Gu et al., 2025) | CogVideoX-5B | I, 3D Track | 0.29 | 15.80 | 0.372 | 0.483 | 40.23 |
| ATI (Wang et al., 2025) | Wan 2.1-14B | I, 2D Track | 0.23 | 15.33 | 0.374 | 0.473 | 17.41 |
| MotionStream Teacher (Shin et al., 2026) | Wan 2.1-1.3B | T, I, 2D Track | 0.79 | **16.61** | 0.477 | 0.427 | 5.35 |
| MotionStream Causal (Shin et al., 2026) | Wan 2.1-1.3B | T, I, 2D Track | 16.7 | 16.20 | 0.447 | 0.443 | 7.80 |
| MotionStream Teacher (Shin et al., 2026) | Wan 2.2-5B | T, I, 2D Track | 0.74 | 16.10 | 0.466 | 0.427 | 7.86 |
| MotionStream Causal (Shin et al., 2026) | Wan 2.2-5B | T, I, 2D Track | 10.4 | 16.30 | 0.456 | 0.438 | 11.18 |
| Our Teacher | Wan 2.2-5B | T, I, 3D Track | 0.84 | 16.04 | **0.479** | **0.404** | **5.29** |
| Our Causal | Wan 2.2-5B | T, I, 3D Track | **20.6** | 15.48 | 0.457 | 0.426 | 5.48 |

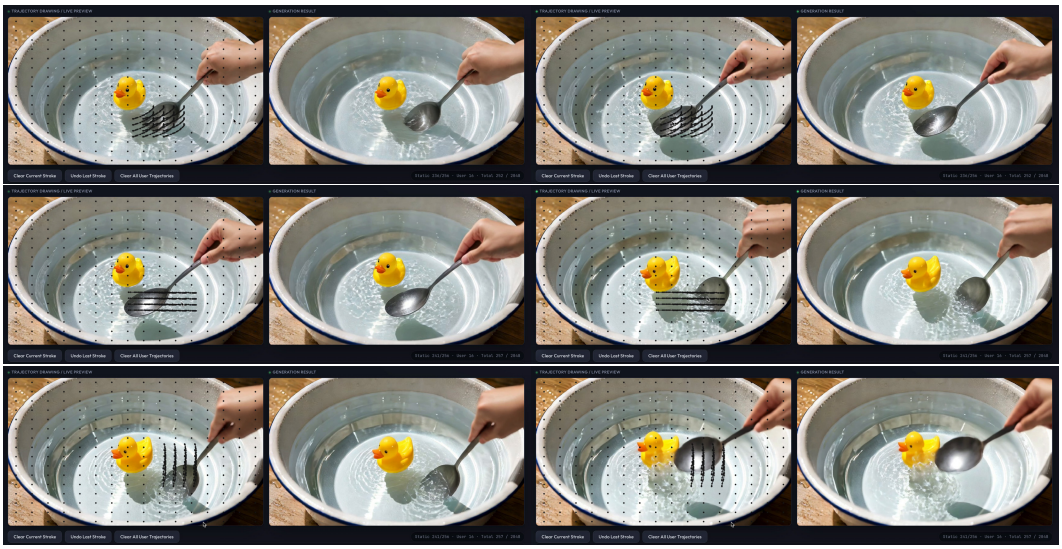


Figure 4: **Frames sampled from a real-time interactive streaming session.** In each panel, the left half shows the user's interaction trajectory and the right half the video frame generated from it. Over a 30-second session the user introduces a spoon to poke a rubber duck, revising the trajectory as the model streams coherent frames that follow the evolving input.

motion head encodes dynamics independently of visual content, so one motion condition can drive diverse target appearances without retraining or per-instance optimization.

### 4.3 Interactive Streaming Control

Our causal student conditions online: track signals arrive sequentially during generation and the model responds within the stream, closing the control loop while the video plays. Fig. 4 samples such a session. Over 30 seconds the user introduces a spoon to interact with a rubber duck, progressively updating the trajectory while the model generates temporally coherent frames that follow the revised input, confirming that the temporally separable motion head supports seamless online conditioning. Extended results up to 350 frames are provided in Section A.1.

The student sustains over 20 FPS at 480p on a single high-end GPU, with memory independent of the generated length. Table 2 summarizes the efficiency gains over the teacher: chunk-wise generation with 4-step denoising yields far lower latency and higher throughput than the teacher's full-sequence 50-step sampling. Peak memory is modestly higher because of the rolling KV cache that preserves cross-chunk consistency, but this footprint is bounded and stays flat as the video lengthens, whereas full-sequence attention grows with clip length. Constant memory *with respect to length*, rather than a lower absolute footprint, is what makes unbounded streaming feasible on one GPU.

Table 2: **Streaming efficiency comparison.** Latency, peak memory, and throughput measured on 20 validation videos. $w$: window size; $c$: chunk size. Best results are in **bold**, second best are underlined.

| Model | Latency (s) ↓ | Memory (GB) ↓ | Throughput (FPS) ↑ |
|---|---|---|---|
| Teacher (Full Attention) | $34.92 \pm 0.351$ | $\mathbf{23.9 \pm 0.82}$ | $0.84 \pm 0.001$ |
| Causal ($w$=21, $c$=4) | $0.756 \pm 0.006$ | $31.8 \pm 0.72$ | $20.6 \pm 0.03$ |
| Causal ($w$=9, $c$=4) | $0.756 \pm 0.006$ | $30.1 \pm 0.65$ | $\mathbf{21.3 \pm 0.01}$ |
| Causal ($w$=21, $c$=1) | $\mathbf{0.411 \pm 0.005}$ | $31.4 \pm 0.68$ | $10.5 \pm 0.02$ |

Table 3: **Ablations of core designs.** "Full" is our complete teacher pipeline. Metrics are reported on 20 clean validation videos with 81 frames from `OpenVidHD-Motion3D`.

| Variant | PSNR↑ | SSIM↑ | LPIPS↓ | EPE↓ |
|---|---|---|---|---|
| Full (3D + ID + 2-stage) | $\mathbf{19.13 \pm 3.16}$ | $\mathbf{0.661 \pm 0.174}$ | $\mathbf{0.200 \pm 0.086}$ | $\mathbf{0.985 \pm 1.095}$ |
| 2D tracks only | $18.95 \pm 3.09$ | $0.649 \pm 0.172$ | $0.225 \pm 0.097$ | $1.523 \pm 1.224$ |
| w/o track-ID embedding | $16.55 \pm 3.32$ | $0.592 \pm 0.180$ | $0.283 \pm 0.090$ | $2.455 \pm 1.148$ |
| Single-stage training (no curriculum) | $17.62 \pm 3.41$ | $0.613 \pm 0.185$ | $0.248 \pm 0.093$ | $3.217 \pm 1.562$ |

### 4.4 Ablation Study

**3D versus 2D motion conditioning.** We compare the full model against a 2D-only variant that drops the depth branch and camera-aware components, conditioning on projected 2D tracks alone under the same motion-head capacity budget. Quantitatively it costs 0.5 EPE and 0.025 LPIPS (Table 3); qualitatively the gap is more telling. As Fig. 5 shows, image-plane conditioning suffers from depth ambiguity and yields physically implausible configurations, while 3D tracks with explicit depth produce motion that respects scene geometry and supports fine-grained control over depth-dependent interactions. This confirms the value of disentangling object motion from camera-induced image motion. Additional comparisons with and without track conditioning appear in Section A.2.

**Motion head.** Removing the track-ID embedding causes the largest single-component drop, with EPE rising from 0.985 to 2.455 and LPIPS from 0.200 to 0.283, indicating that sinusoidal identity encoding is what lets the model tell individual trajectories apart and follow each faithfully.

**Training curriculum.** Single-stage training at full resolution degrades every metric (EPE 3.217 versus 0.985, SSIM 0.613 versus 0.661), confirming that the low-resolution warm-up stabilizes motion alignment before scaling to higher resolution and longer horizons. Further ablations on distillation hyperparameters, random seeds, and the number of control points are provided in Sections C.1 to C.3.

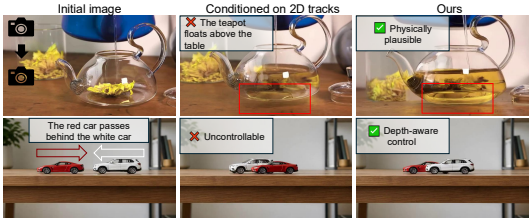



Figure 5: **Advantage of 3D over 2D track conditioning.** *Top:* under a camera move-down, 2D tracks leave the teapot floating above the table through depth ambiguity, whereas our 3D-aware conditioning keeps it grounded. *Bottom:* instructed that the red car should pass behind the white car, 2D conditioning resolves the depth ordering incorrectly, while our depth-aware control respects the intended occlusion.

## 5 Conclusion

We presented 4DStreamCtrl, a controllable video generation framework that unifies camera motion, object trajectories, and per-point depth into a single 3D-track interface for 4D control. With large-scale in-the-wild 3D motion supervision, a temporally separable Geometric Motion Head, and a causal streaming distillation, it achieves state-of-the-art motion-control precision while enabling, to our knowledge, the first real-time 4D-controllable streaming video synthesis on a single GPU. These results suggest that grounding generative video in explicit 3D geometry, coupled with efficient causal inference, is a viable path toward interactive world models with closed-loop spatiotemporal control. We discuss remaining failure modes in Section A.3 and outline limitations and future directions in Section D.

## REFERENCES


Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. $\pi_0$: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. A9

Jake Bruce, Michael D. Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Maria Elisabeth Bechtle, Feryal Behbahani, Stephanie C.Y. Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando De Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *International Conference on Machine Learning (ICML)*, 2024. 3, A9

Ryan Burgert, Yuancheng Xu, Wenqi Xian, Oliver Pilarski, Pascal Clausen, Mingming He, Li Ma, Yitong Deng, Lingxiao Li, Mohsen Mousavi, Michael Ryoo, Paul Debevec, and Ning Yu. Go-with-the-flow: Motion-controllable video diffusion models using real-time warped noise. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2, 6, 8

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 (NeurIPS)*, 2024. 2, 5

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Joshua B Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. A9

Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Yusuf Aytar, Michael Rubinstein, Chen Sun, et al. Motion prompting: Controlling video generation with motion trajectories. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 2, 3

Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, Wenping Wang, and Yuan Liu. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In *ACM SIGGRAPH Conference Proceedings*, 2025. 2, 3, 6, 8

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. A9

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024. 2

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 3

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. 5, A3

Yuanming Hu, Luke Anderson, Tzu-Mao Li, Qi Sun, Nathan Carr, Jonathan Ragan-Kelley, and Frédo Durand. Difftaichi: Differentiable programming for physical simulation. In *International Conference on Learning Representations (ICLR)*, 2020. A9

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 (NeurIPS)*, 2025. 2, 3, 5, A5

Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker3: Simpler and better point tracking by pseudo-labelling real videos. In *International Conference on Computer Vision (ICCV)*, 2025. 3, A9

Dan Kondratyuk, Lijun Yu, Xiuye Gu, José 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 *International Conference on Machine Learning (ICML)*, 2024. 3

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024. 2

Yao-Chih Lee, Zhoutong Zhang, Jiahui Huang, Jui-Hsien Wang, Joon-Young Lee, Jia-Bin Huang, Eli Shechtman, and Zhengqi Li. Generative video motion editing with 3d point tracks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026. 2

Yaowei Li, Xintao Wang, Zhaoyang Zhang, Zhouxia Wang, Ziyang Yuan, Liangbin Xie, Ying Shan, and Yuexian Zou. Image conductor: Precision control for interactive video synthesis. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2025. 2, 3, 6, 8

Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *International Conference on Learning Representations (ICLR)*, 2019. A9

Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. In *International Conference on Learning Representations (ICLR)*, 2025. 2, 3

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *International Conference on Computer Vision (ICCV)*, 2023. 2, 3

Federico Perazzi, Jordi Pont-Tuset, Brian McWilliams, Luc Van Gool, Markus Gross, and Alexander Sorkine-Hornung. A benchmark dataset and evaluation methodology for video object segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations (ICLR)*, 2022. 3

Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3, A9

Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *International Conference on Learning Representations (ICLR)*, 2024. A9

Joonghyuk Shin, Zhengqi Li, Richard Zhang, Jun-Yan Zhu, Jaesik Park, Eli Shechtman, and Xun Huang. Motionstream: Real-time video generation with interactive motion controls. In *International Conference on Learning Representations (ICLR)*, 2026. 2, 3, 4, 5, 6, 8

Yao Teng, Enze Xie, Yue Wu, Haoyu Han, Zhenguo Li, and Xihui Liu. Drag-a-video: Non-rigid video editing with point-based interaction. *arXiv preprint arXiv:2312.02936*, 2023. 2, 3

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 2, 3, 4, 5, A3

Angtian Wang, Haibin Huang, Zhiyuan Fang, Yiding Yang, and Chongyang Ma. Ati: Any trajectory instruction for controllable video generation. *arXiv preprint arXiv:2505.22944*, 2025. 2, 3, 6, 8

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jérôme Revaud. Dust3r: Geometric 3d vision made easy. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024a. 3, A9

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH Conference Proceedings*, 2024b. 2, 3

Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024a. A9

Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. Draganything: Motion control for anything using entity representation. In *European Conference on Computer Vision (ECCV)*, 2024b. 2, 3

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024. 2, 3, A9

Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: 3d point tracking made easy. In *International Conference on Computer Vision (ICCV)*, 2025. 2, 4, A4, A8

Sherry Yang, Yilun Du, Kamyar Ghasemipour, Jonathan Tompson, Leslie Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations (ICLR)*, 2024. 3, A9

Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and William T Freeman. Improved distribution matching distillation for fast image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a. 2, 3

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Frédo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024b. 2, 3, 6, A5

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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 3, 6, A5

Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *International Conference on Computer Vision (ICCV)*, 2023. A9

Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G. Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, and Lu Jiang. Magvit: Masked generative video transformer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 6

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Re, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. A9

# APPENDIX

**A More Visualization** **A1**
A.1 Long Video Generation . . . . . A1
A.2 With or Without Track Conditioning . . . . . A1
A.3 Failure Cases . . . . . A2

**B Training Details** **A3**
B.1 Teacher Training . . . . . A3
B.2 Streaming Distillation . . . . . A5

**C More Ablations** **A6**
C.1 Ablations on Distillation . . . . . A6
C.2 Effect of Random Seeds . . . . . A7
C.3 Effect of the Number of Control Points . . . . . A7
C.4 Effect of Attention Sink . . . . . A8

**D Limitations and Future Work** **A8**

## A MORE VISUALIZATION

### A.1 LONG VIDEO GENERATION

We demonstrate generation well beyond the training clip length. Fig. A1 shows three sequences spanning up to 350 frames: a seated woman gesturing at a table (top), a woman cooking in a kitchen (middle), and a close-up of hands performing delicate craft work (bottom).

Across all three, appearance stays plausible and motion temporally coherent for the full duration, with fine-grained detail such as facial features, hand movement, and object interaction remaining consistent into the later frames. Per-frame SSIM starts high (*e.g.*, above 0.90) and declines gradually as the sequence progresses, which is expected given the growing temporal distance from the conditioning frame, yet stays above 0.75 throughout, indicating that visual fidelity is well preserved.

Track accuracy stays low for most frames, with occasional spikes at rapid motion events. The third sequence shows a pronounced EPE peak around frame 125, coinciding with a fast hand movement near the sewing machine. These spikes are transient and the metric recovers quickly, so abrupt motion does not cascade into persistent drift. Together these results confirm that our approach extends to several hundred frames while maintaining quality in both appearance and motion fidelity.

### A.2 WITH OR WITHOUT TRACK CONDITIONING

We qualitatively examine how 3D point track conditioning affects generation on two representative scenarios, shown in Fig. A2.

**Rolling ball.** Without track conditioning (first group), the model produces plausible initial frames but progressively enlarges the ball, violating size constancy. With track conditioning (second group), the trajectories anchor the ball's spatial extent across frames, holding scale constant and producing a natural rolling motion.

**Pushing a can.** Without track guidance (third group), the pushed can undergoes implausible rotation and deformation as the sequence progresses, hallucinating motion incoherent with the initial contact. Conditioning on point tracks (fourth group) constrains the can to a physically plausible rigid-body trajectory, preserving shape and orientation while reflecting the applied force direction.




Figure A1: **Long video generation.** Three sequences generated by our method, each shown as six uniformly sampled frames (left) with per-frame SSIM ↑ and EPE ↓ over the full sequence (right).

**Discussion.** Text prompts alone provide insufficient geometric constraint for fine-grained object dynamics. Point track conditioning acts as a spatial prior that fixes *where* objects should be at each timestep, complementing the semantic guidance from text that specifies *what* should happen. The combination resolves common failure modes of unconditioned generation, including object drift, scale inconsistency, and implausible deformation, without explicit physics simulation.

### A.3 Failure Cases

We identify two representative failure modes. First, when subjects or objects occupy a small region of the frame, results tend to blur, losing fine detail such as facial features and object texture (Fig. A3, top). This likely follows from the limited spatial resolution allocated to small regions, which makes high-frequency detail hard to sustain over time. Second, our method can produce sequences that violate causality and object permanence (Fig. A3, bottom): after a topping is placed on a cake, the cake inexplicably develops a missing chunk, and a piping bag appears from nowhere in later frames. These artifacts suggest the model lacks a coherent account of causal object interaction and struggles to hold object state consistent across frames. Incorporating structured physical or causal reasoning is a promising direction for future work.

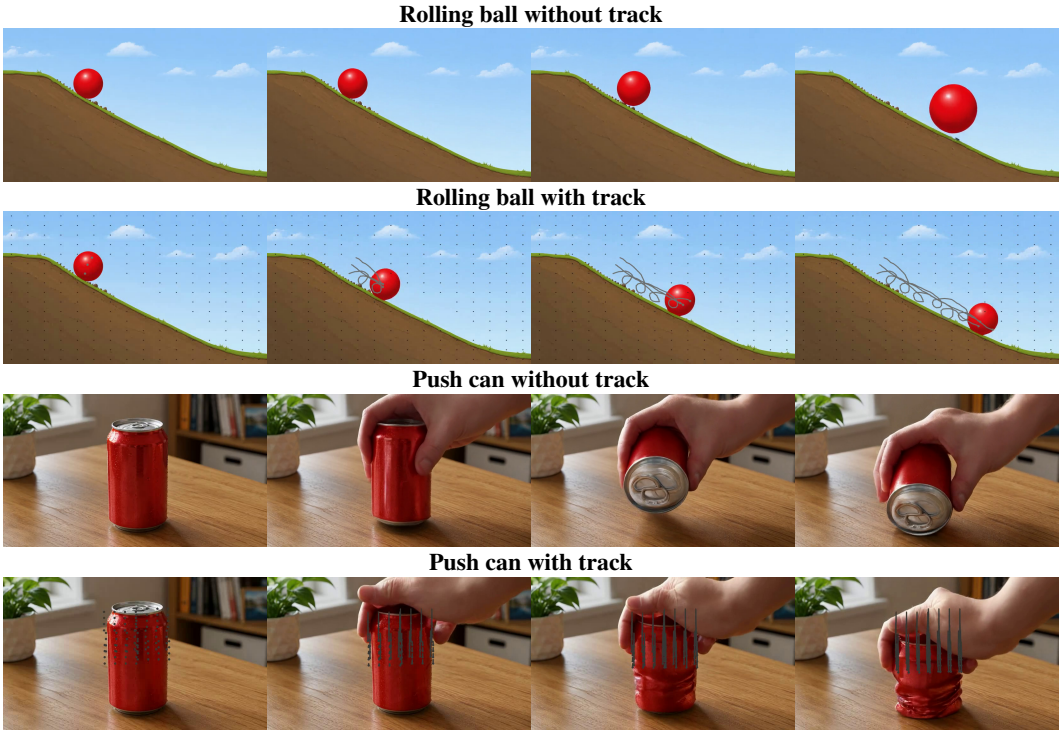



Figure A2: **Effect of point track conditioning.** Without track guidance the rolling ball grows unnaturally over time and the pushed can rotates implausibly. Track conditioning enforces spatial consistency, preserving object scale and yielding more physically coherent interaction.

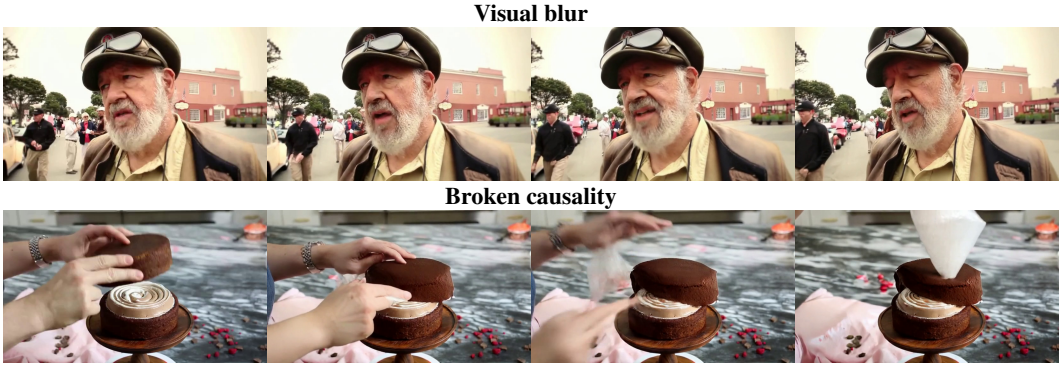



Figure A3: **Failure cases.** Top: small faces and background objects blur progressively as the video extends. Bottom: causally inconsistent interaction, where a cake develops an unexplained missing section after being topped and a piping bag materializes from thin air.

# B Training Details

## B.1 Teacher Training

**Base model and fine-tuning strategy.** The teacher is the Wan2.2 text-and-image-to-video DiT with 5B parameters (Wan et al., 2025), operating on a latent space from a causal 3D VAE with spatial compression $16\times$ and temporal compression $4\times$. It takes 48-channel video latents and generates 81-frame clips at $480 \times 832$. We freeze the pretrained DiT and inject LoRA adapters (Hu et al., 2022) of rank $r$=32 into the query, key, value, and output projections (`q, k, v, o`) and both FFN linear layers (`ffn.0, ffn.2`). Alongside the LoRA parameters we jointly train:

- the motion track encoder;
- the expanded patch embedding, whose input channels grow from $C_{\text{lat}}$=48 to $C_{\text{lat}} + C_{\text{trk}} + C_{\text{dep}} = 48 + 64 + 16 = 128$ to accept the concatenated motion features.

Table A1: **Teacher model training configuration.**

| Hyperparameter | Value |
|---|---|
| Base model | Wan2.2-TI2V-5B |
| Resolution (stage 1) | $256 \times 256$, 49 frames |
| Resolution (stage 2) | $480 \times 832$, 81 frames (resumed from stage 1) |
| LoRA rank | 32 |
| LoRA target modules | `q, k, v, o, ffn.0, ffn.2` |
| Learning rate (stage 1) | $1 \times 10^{-4}$ |
| Learning rate (stage 2) | $5 \times 10^{-5}$ |
| Epochs | 3 |
| GPUs | 80 (multi-node, NCCL) |
| Precision | BF16 (mixed precision) |
| Gradient checkpointing | Enabled |
| Track embed dim $D$ | 64 |
| Track head output channels $C_{\text{trk}}$ | 64 |
| Depth embed dim $D_d$ | 16 |
| Depth head output channels $C_{\text{dep}}$ | 16 |
| Tracks per sample $N$ | 256 (fixed) |
| Max track ID | 256 |
| Dataset | `OpenVidHD-Motion3D` |
| Data workers | 8 per process |
| Distributed framework | Accelerate (DDP) |

The original patch embedding weights are preserved and the new channels zero-filled, giving a stable initialization from pretrained knowledge. The conditioning tensor $\mathbf{F}^{\text{cond}} = [\mathbf{F}^{\text{trk}}; \mathbf{F}^{\text{dep}}] \in \mathbb{R}^{80\times F\times H_\ell \times W_\ell}$ is concatenated with the noisy latents along the channel dimension before patch embedding, as defined in Section 3.2.

**3D track data preparation.** We extract per-video 3D tracks with SpatialTrackerV2 (Xiao et al., 2025) and store them in a single HDF5 file with per-video groups, each containing:

- `tracks`: $[N, T, 3]$ tensor of $(x, y, z)$ in camera space;
- `intrinsics`: $[T, 3, 3]$ camera intrinsic matrices;
- `visibility`: $[N, T]$ binary occlusion mask, not used in training; occlusion and missing samples are handled implicitly by the depth branch and by empty raster cells.

At load time, tracks are projected to image coordinates with the intrinsics, giving normalized positions in $[0, 1]^2$. Depth is converted to inverse depth and min–max normalized to $[0, 1]$ per sample. Each training sample draws a fixed $N$=256 tracks by random permutation of the tracked points in that video.

**Training objective.** We adopt the rectified flow-matching objective of Wan2.2. Given clean latents $\mathbf{x}_0$ encoded from the training video, Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and a flow time $\tau \sim \mathcal{U}(0, 1000)$ warped by the shifted-sigmoid schedule (shift $s$=5, $\sigma(\tau) = s\tau/(1 + (s - 1)\tau)$), the noisy latent is

$$\mathbf{x}_\tau = (1 - \sigma_\tau)\, \mathbf{x}_0 + \sigma_\tau\, \boldsymbol{\epsilon}. \tag{A1}$$

The velocity target is $\mathbf{v} = \boldsymbol{\epsilon} - \mathbf{x}_0$, and we minimize

$$\mathcal{L} = \left\| \mathbf{v}_\theta(\mathbf{x}_\tau, \tau, \mathbf{c}, \mathbf{F}^{\text{cond}}) - \mathbf{v} \right\|_2^2, \tag{A2}$$

where $\mathbf{c}$ denotes text and image conditioning. The first-frame latent is injected as ground truth rather than noised, and the loss is evaluated only on the remaining latent frames.

**Training configuration.** Table A1 summarizes the key hyperparameters. Training uses the Accelerate framework with DDP across multiple nodes, and gradient checkpointing throughout the DiT backbone to fit 81-frame clips within per-GPU memory. Trainable parameters comprise LoRA adapters in all 40 transformer blocks (about 200M at rank 32), the motion track encoder (about 0.3M), the depth branch (about 0.1M), and the expanded patch embedding (about 2.6M for the 80 new input channels).

Table A2: **ODE initialization training configuration.**

| Hyperparameter | Value |
|---|---|
| Student architecture | Wan2.2-TI2V-5B (causal, block-wise attention) |
| Frames per chunk $c$ | 4 |
| Denoising steps (warped) | $\{1000, 750, 500, 250\}$ |
| Independent first frame | Yes |
| Learning rate | $2 \times 10^{-5}$ |
| Optimizer | AdamW ($\beta_1$=0.9, $\beta_2$=0.999, wd=0.01) |
| Global batch size | 64 |
| Per-GPU batch size | 1 |
| Precision | BF16 (mixed) |
| Gradient checkpointing | Enabled |
| FSDP strategy | Hybrid full shard |
| Dataset | Pre-computed ODE pairs (LMDB) |
| Motion track channels | $C_{\text{trk}}$=64, $C_{\text{dep}}$=16 |
| Tracks per sample | 256 |

### B.2 Streaming Distillation

We distill the bidirectional teacher into a causal 4-step student following the self-forcing and DMD paradigm (Huang et al., 2025; Yin et al., 2024b; 2025), in three sequential stages: ODE pair generation, ODE initialization, and adversarial DMD training.

**Stage 0: ODE pair generation.** We run the frozen teacher for 48 denoising steps (shifted-sigmoid schedule, shift $s$=5, $\sigma_{\min}$=0) with classifier-free guidance at scale 1 on each training video to produce full ODE trajectories. We use 48 rather than the default 50 steps so the trajectory divides evenly into the student's 4-step schedule, giving clean intermediate snapshots at uniform intervals. The teacher receives the first-frame latent, held fixed at frame 0 throughout, together with 3D motion track conditioning. We save 5 evenly spaced snapshots per trajectory (steps 0, 12, 24, 36, and the final clean latent), yielding tensors of shape $[5, F, C_{\text{in}}, H_\ell, W_\ell]$ with $F$=21, $C_{\text{in}}$=48+80, $H_\ell$=30, and $W_\ell$=52. Snapshots are aggregated into an LMDB database for fast random access during training.

**Stage 1: ODE initialization.** We initialize the causal student by supervising it on the precomputed ODE pairs. The student shares the teacher's architecture but replaces bidirectional self-attention with **block-wise causal attention**: latent frames are grouped into chunks of $c$=4 frames, and when denoising chunk $i$ the model attends only to its own tokens and a causally accumulated KV cache from preceding chunks. At each step we sample an intermediate denoising index per block, extract the corresponding noisy latent from the saved trajectory, and regress the clean latent,

$$\mathcal{L}_{\text{ode}} = \left\| \hat{\mathbf{x}}_{0,\theta}(\mathbf{x}_{\tau_k}, \tau_k, \mathbf{c}) - \mathbf{x}_0 \right\|_2^2, \tag{A3}$$

with $\tau_k \in \{1000, 750, 500, 250\}$ after warping by the shifted schedule. The first-frame latent is injected as ground truth and excluded from the loss. Table A2 lists the hyperparameters.

**Stage 2: DMD adversarial distillation.** We then switch to distribution matching distillation (Yin et al., 2024b) with temporal self-rollout, using the objective given in Section 3.2. Training involves three models:

- **Generator** $G_\theta$, the causal student, initialized from the ODE stage checkpoint.
- **Real score** $s_{\text{real}}$, the frozen LoRA-finetuned bidirectional teacher loaded via DiffSynth with motion track conditioning.
- **Fake score** $s_{\text{fake}}$, a learnable critic initialized from the same teacher weights and trained online.

At each generator step the student produces a full clip by temporal self-rollout: every chunk of $c$=4 frames is denoised through the 4-step schedule $\{1000, 750, 500, 250\}$ conditioned on its own previously generated latents via KV cache, matching streaming deployment, with the same denoising step used across all chunks within an iteration. The real score applies classifier-free guidance at scale 1 while the fake score uses the conditional prediction alone. The critic is updated at every step and the generator once every 5 critic steps, with an exponential moving average of the generator weights at decay 0.99 from step 200 onward. Table A3 lists the full configuration.

Table A3: **DMD distillation training configuration.**

| Hyperparameter | Value |
|---|---|
| Generator init | ODE stage checkpoint |
| Real score (teacher) | Frozen LoRA-finetuned Wan2.2-TI2V-5B |
| Fake score (critic) init | Same teacher weights (trainable) |
| LoRA alpha | 1.0 |
| CFG scale (teacher) | 1.0 |
| Denoising steps | $\{1000, 750, 500, 250\}$ (4 steps) |
| Frames per chunk $c$ | 4 |
| Timestep shift $s$ | 5.0 |
| Timestep range | $[20, 980]$ |
| Context noise | 0 (clean KV cache) |
| Generator lr | $2 \times 10^{-6}$ |
| Critic lr | $4 \times 10^{-6}$ |
| Optimizer | AdamW ($\beta_1$=0, $\beta_2$=0.999) |
| Global batch size | 64 |
| Per-GPU batch size | 1 |
| Critic-to-generator update ratio $R$ | 5 |
| EMA decay | 0.99 (start step 200) |
| Max gradient norm | 10.0 (both) |
| Precision | BF16 (mixed) |
| Gradient checkpointing | Enabled |
| FSDP strategy | Hybrid full shard |
| Dataset | `OpenVidHD-Motion3D` |
| Resolution | $480 \times 832$, 81 frames |
| Latent shape | $[1, 21, 48, 30, 52]$ |
| Tracks per sample | 256 |
| Denoising loss type | Flow matching |

**Distributed training.** Both stages use multi-node training with fully sharded data parallelism (FSDP, hybrid full-shard). The generator, real score, fake score, and text encoder are independently FSDP-wrapped with size-based auto-wrapping, while the VAE remains unsharded on each device for online latent encoding during DMD training. Distillation runs on 64 high-end GPUs.

**Online 3D control at inference.** Because the motion encoder is temporally separable (Section 3.2), switching between offline full-sequence and online per-chunk conditioning requires no architectural change. Generation proceeds chunk by chunk: once a chunk is committed and decoded, the encoder recomputes $\mathbf{F}^{\text{cond}}$ from the latest 3D tracks and depth over the upcoming segment. Users may revise trajectories, supply new tracker outputs, or adjust depth at any time, since only the current control window is required. This closes the loop between interactive 4D control and real-time video output at a memory cost independent of length.

# C More Ablations

## C.1 Ablations on Distillation

We ablate two distillation hyperparameters: the classifier-free guidance scale (cfg $\in \{1, 2, 3\}$) and the learning rate (lr $\in \{$3e-6, 4e-6$\}$), tracking PSNR ↑, SSIM ↑, LPIPS ↓, and EPE ↓ over training up to 4K steps (Fig. A4).

All configurations improve steadily on the appearance metrics as training progresses, confirming that distillation works as intended. Among the guidance scales, cfg = 1 is best on all four metrics, reaching the highest PSNR (about 18.5) and SSIM (about 0.65) and the lowest LPIPS (about 0.22) and EPE (about 0.95) at 4K steps. Raising guidance to 2 or 3 degrades results noticeably, suggesting that stronger guidance introduces artifacts during distillation that harm both visual quality and motion accuracy.

For the learning rate, lr = 4e-6 (dashed) consistently beats lr = 3e-6 (solid) across every guidance scale and metric, indicating that a moderately higher rate accelerates convergence without instability. The margin is widest on PSNR and LPIPS, where the dashed curves stay clearly ahead throughout.

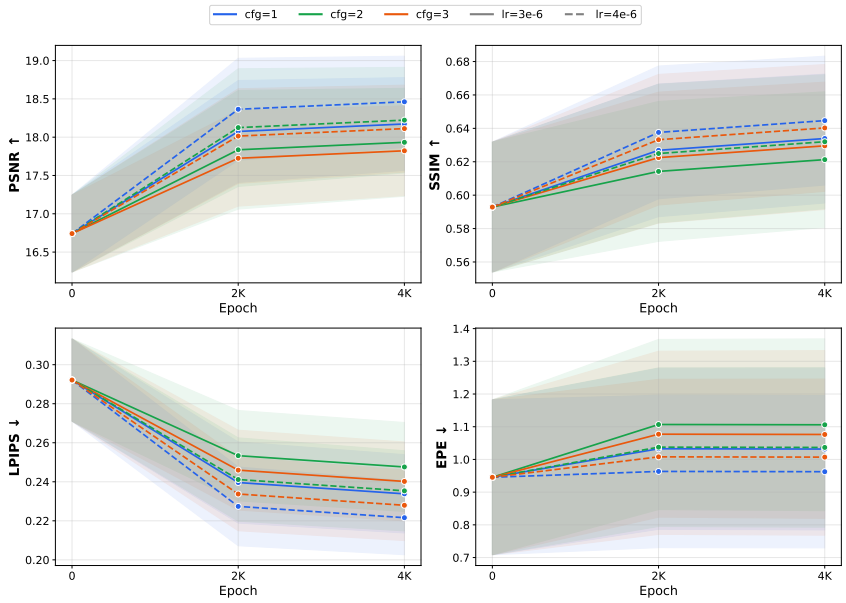



Figure A4: **Distillation ablation.** PSNR, SSIM, LPIPS, and EPE over 4K distillation steps for three guidance scales and two learning rates. Solid curves are lr = 3e-6 and dashed curves lr = 4e-6.

EPE behaves differently from the appearance metrics: cfg = 1 keeps it low and stable, whereas cfg = 2 and cfg = 3 drift slightly upward, so higher guidance appears to introduce temporal inconsistencies that accumulate over longer training. We therefore adopt cfg = 1 with lr = 4e-6 as the default.

### C.2 Effect of Random Seeds

To probe the stochasticity of generation in our distilled student model, we fix the first frame and the motion trajectories and vary only the random seed. Fig. A5 shows five videos generated with seeds 0 to 4 under identical conditioning. Regions governed directly by the input trajectories move almost identically across seeds, indicating that trajectory control is a strong and reliable spatial signal.

Variation instead concentrates where the conditioning is ambiguous or underspecified. Newly entering objects such as hands, whose appearance and timing the trajectories do not dictate, differ across seeds in shape, pose, and moment of entry. Occluded regions such as the interior of the cooking pan likewise settle into different fine-grained states, for instance the arrangement and appearance of the food, since neither the first frame nor the trajectories constrain them. This behavior is both expected and desirable: the model follows the provided control signals faithfully while using the seed to sample plausible completions for the remaining degrees of freedom, producing outputs that are diverse yet coherent.

### C.3 Effect of the Number of Control Points

We study how the number of control tracks used at student inference affects generation quality and motion following, varying it over $\{16, 32, 64, 128, 256, 512, 1024\}$ with all other settings fixed (Fig. A6).

From 16 to 256 tracks, quality improves consistently: PSNR rises from 16.25 to 18.27, SSIM from 0.55 to 0.64, and LPIPS falls from 0.29 to 0.23, so denser control points provide richer spatial guidance. EPE stays stable across this range, fluctuating between 2.60 and 2.75.

Beyond 256 tracks all metrics degrade, moderately at 512 (PSNR 18.02, SSIM 0.62) and sharply at 1024 (PSNR 16.21, LPIPS 0.31, EPE 3.41). Since the model is trained with a fixed 256 tracks, this is expected: inference-time counts far from the training distribution give suboptimal results. The degradation is more severe above the training count than below it, as the model handles sparser subsets of the learned conditioning pattern more gracefully. We therefore use 256 tracks by default in all other experiments, matching the training configuration and giving the best balance of quality and motion accuracy.

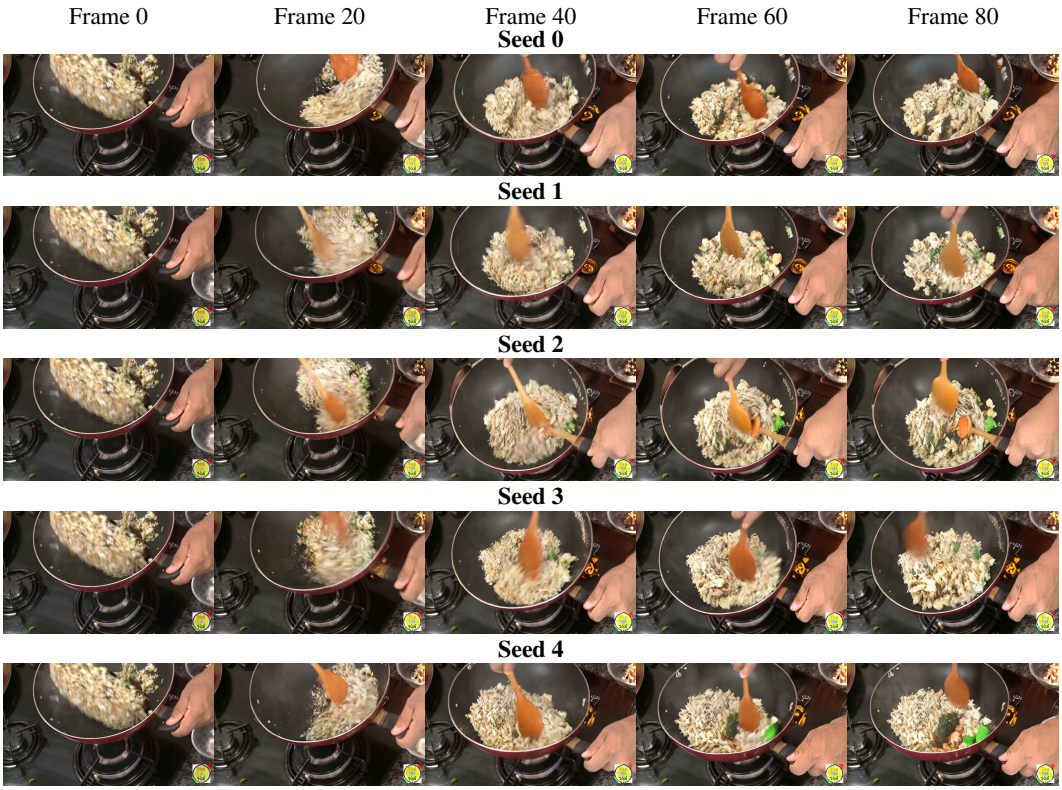



Figure A5: **Generated frames across random seeds.** All five sequences share the same first frame and 3D tracks and differ only in seed. Trajectory-governed regions stay consistent, while underspecified content such as entering hands and occluded interiors varies.

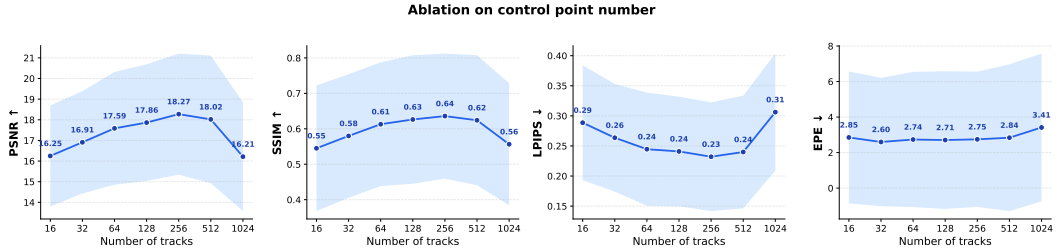



Figure A6: **Effect of the number of control points on student model generation.** PSNR, SSIM, LPIPS, and EPE as the inference-time track count varies from 16 to 1024. Quality peaks at the training count of 256.

### C.4 Effect of Attention Sink

Our autoregressive generation uses a sliding context window, so earlier tokens are progressively evicted from the KV cache as the sequence extends. To keep a stable reference, we retain the initial token as a persistent anchor throughout generation.

Fig. A7 shows the effect. Without the sink, frames drift cumulatively: color saturation intensifies, facial identity departs from the initial appearance, and fine detail degrades into an overly smooth rendering. With the sink retained, the model holds color, identity, and visual quality stable across the entire sequence.

## D Limitations and Future Work

**Dependency on monocular 3D estimation.** Our data pipeline relies on monocular 3D point tracking (Xiao et al., 2025) for scalable supervision without calibrated sensors. This enables large-scale curation from internet video, but the resulting 3D labels carry inherent scale ambiguity and grow less reliable under rapid motion or severe occlusion. Data diversity absorbs most of this during training, though it can surface at inference as subtle depth-scale inconsistency under extreme camera base-




Figure A7: **Effect of the attention sink.** Frames sampled at equal intervals from a 350-frame generated video. Without the sink, color and identity drift as the sequence extends.

lines. Adding multi-view geometric constraints (Schönberger & Frahm, 2016; Wang et al., 2024a) or metric depth priors (Yin et al., 2023) during curation could sharpen the supervision.

**Fixed-density trajectory representation.** Our $32 \times 32$ track grid balances compute against spatial coverage for most scenarios. Scenes with highly localized fine-grained interaction, such as dexterous hand manipulation or dense multi-body contact, would benefit from spatially adaptive track allocation. Extending the interface to variable-density or hierarchical layouts is a natural direction that could improve controllability in such cases without raising the average budget.

**Streaming student versus teacher gap.** The causal student achieves a $12.5\times$ speedup with only modest quality loss relative to the bidirectional teacher (about 0.5 dB PSNR and a marginal EPE increase on DAVIS, Table 1). The fixed local attention window suffices for most motion patterns in our evaluation, though dependencies reaching well beyond it could be captured better with adaptive memory (Xiao et al., 2024; Zhang et al., 2023). We view closing this gap as an engineering challenge rather than a fundamental limit of the framework.

**Future directions.** Building on the current system, we see several promising extensions:

- **Closed-loop perception and generation.** Integrating real-time visual trackers (Karaev et al., 2025) that observe the generated output and feed corrective 3D tracks back into the conditioning pipeline would give self-correcting generation, analogous to model-predictive control, compensating for drift without user intervention.
- **Physics-informed trajectory synthesis.** Coupling our interface with differentiable physics engines (Hu et al., 2020) or learned dynamics models (Li et al., 2019) could generate physically plausible trajectories directly from high-level intent, reducing the manual effort of specifying control signals.
- **Embodied AI deployment.** Our streaming architecture interfaces naturally with robotic planning stacks, where action-conditioned 3D motion goals must be rendered as photorealistic futures for decision-making (Du et al., 2023; Yang et al., 2024; Black et al., 2024). Extending 4DStreamCtrl into a real-time visual imagination module for model-based policy learning is a compelling direction.
- **4D-consistent multi-view generation.** Our 3D track conditioning enforces single-view geometric consistency; combining it with multi-view diffusion (Shi et al., 2024) or 4D scene representations (Wu et al., 2024a) could yield generators with strict cross-view consistency, enabling VR and AR content creation and digital twin synthesis.
- **Semantic trajectory understanding.** Trajectories currently encode pure kinematics without causal semantics. Incorporating world-model reasoning (Bruce et al., 2024; Ha & Schmidhuber, 2018) that couples "where to move" with "why to move" could improve physical plausibility and enable intent-level control beyond explicit point specification.