TRACK2VIEW: 4D-CONSISTENT CAMERA-CONTROLLED VIDEO GENERATION VIA PAIRED 3D POINT TRACKS

Anonymous authors
Paper under double-blind review

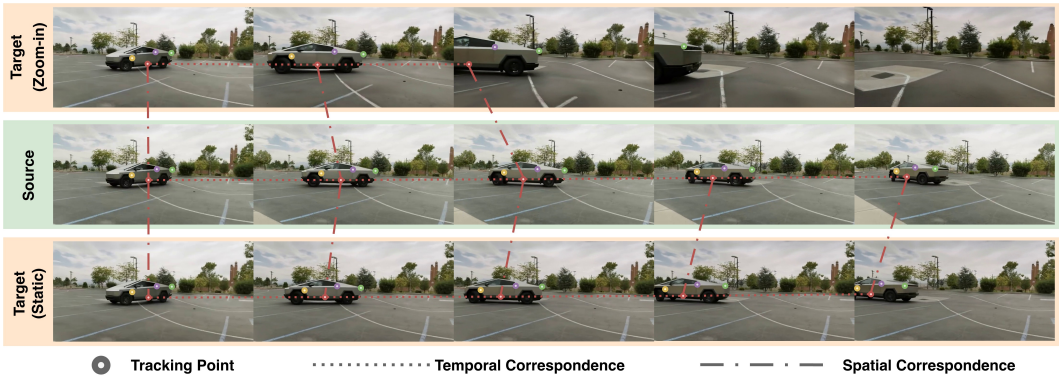

Figure 1: **Track2View** re-renders a single source video under arbitrary target camera trajectories by conditioning on sparse 3D point tracks. Colored markers denote tracked object points; scene-point tracks are also used for conditioning but omitted from the visualization for clarity. *Spatial correspondence* lines link the same point across views at a fixed timestep, while *temporal correspondence* lines link it across frames within a view. Together, these correspondences expose the 4D consistency that Track2View enforces between source and generated views.

ABSTRACT

Re-rendering an existing video from a novel camera viewpoint requires the output to follow the prescribed camera trajectory while preserving the appearance and dynamics of the original scene across every frame. Existing methods either inject camera parameters and leave source-to-target correspondence implicit, or render estimated geometry frame by frame; neither provides an explicit, temporally continuous link between source and target pixels. We propose Track2View, which conditions a video diffusion transformer on paired 3D point tracks: sparse trajectories of scene points projected into both the source and target camera views. These tracks provide explicit spatiotemporal correspondences that are temporally continuous by construction, encoding what content should appear where and when. At the core of Track2View is a dual-view track conditioner that transfers visual context from source to target view through parameter-free geometric operations and learned temporal aggregation, ensuring generalization to arbitrary camera trajectories without memorizing specific motions. We further introduce a data curation pipeline that extracts one-to-one track correspondences by running a 3D point tracker on temporally concatenated multi-camera view pairs. On a 400-video benchmark spanning static and dynamic scenes, Track2View achieves state-of-the-art results across visual quality, view synchronization, and camera accuracy, reducing rotation error by 20–39% and translation error by 58–66% relative to leading baselines.

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

Controlling camera viewpoints in video generation is essential for applications ranging from filmmaking and virtual reality to robotics simulation. Recent video diffusion models have enabled high-quality text-to-video and image-to-video synthesis (Blattmann et al., 2023; Wan et al., 2025; Yang et al., 2025; Agarwal et al., 2025), yet precisely steering the camera of a generated video remains a fundamental challenge. The difficulty is amplified in the video-to-video (V2V) setting, where an existing video is re-rendered along a new camera trajectory: each output frame must follow the prescribed camera while reproducing the appearance and dynamics of the source video at the same timestep. This task demands 4D consistency—coherence across viewpoints and across time at once—and therefore depends on how the generator is told which source content belongs where in every target frame.

Existing camera-controlled V2V methods differ mainly in how geometry enters the generator. Camera-conditioned methods such as ReCamMaster (Bai et al., 2025) inject a learned camera embedding and leave every source-to-target correspondence to be inferred by attention. Geometry-rendering methods such as Gen3C (Ren et al., 2025) and TrajectoryCrafter (Yu et al., 2025) lift the source video to a point cloud with monocular depth and render it into each target view, and Trajectory Attention (Xiao et al., 2025b) conditions on pixel trajectories obtained by the same kind of reprojection; the latter two also define the target camera relative to each source frame, implicitly assuming a static source camera. Rendered guidance makes correspondence explicit, but it is a frame-local projection: the guidance for each target frame is rendered only from the corresponding source frame, so it contains nothing that is hidden in that frame, and every depth error becomes a dense region of incorrect pixels.

We argue that the condition should state explicitly which source content belongs where, in every frame and across time, and that paired 3D point tracks provide exactly this. When the same 3D points are tracked through the source video and projected under the target trajectory, each track links a source pixel to its target location at every timestep (Fig. 1). Compared with rendered geometry, paired tracks have three properties. First, they preserve point identity: a point that is occluded and later reappears remains the same track, so its appearance can be gathered from every frame in which it is visible rather than only from the current source frame. Second, they are defined in a global coordinate system and remain valid under arbitrary source camera motion. Third, they are sparse and selectable: the condition is a set of correspondences that can be subsampled or filtered, rather than a dense image that contains every estimated depth value. In controlled comparisons, paired tracks outperform a warped-RGB condition built from ground-truth depth, and a small set of confident tracks is more accurate than the full dense set (App. D and E). Tracks are still estimated from the source video, so upstream errors do propagate, but they enter the condition as a limited set of correspondences rather than as densely rendered pixels.

Point tracks have been used as control signals before (Sec. 2.3), so what matters is how they are used. DaS (Gu et al., 2025) renders 3D tracks into a tracking video that conditions an image-to-video model; its appearance comes from the first frame alone, and the tracks act as rendered colors rather than as a channel for source content. Edit-by-Track (Lee et al., 2025) conditions V2V generation on 3D tracks for motion editing; its target tracks come from an edited point cloud, so they are related to the source through learned cross-attention. In Track2View, source and target tracks are the same 3D points by construction: during training, our curation pipeline tracks through temporally reversed and concatenated view pairs, and at inference the target tracks are reprojections of the source tracks. This one-to-one pairing makes a parameter-free transfer possible: the conditioner samples source-video features along each track, scatters them to the paired target locations by bilinear interpolation, and learns only how to aggregate each track over time.

Our main contributions are as follows:

- We propose Track2View, which conditions a video diffusion transformer on paired 3D point tracks through a dual-view track conditioner that transfers source-video features to target locations with parameter-free bilinear sampling and scattering, while learning only temporal aggregation.
- We introduce a data curation pipeline that obtains one-to-one paired tracks from synchronized multi-camera videos by tracking through temporally concatenated view pairs; this pairing by construction is what allows the geometric transfer to be parameter-free. We will release the resulting track annotations.

- On a 400-video benchmark spanning static and dynamic scenes, Track2View achieves state-of-the-art results across visual quality, view synchronization, and camera accuracy, reducing rotation error by 20–39% and translation error by 58–66% relative to the strongest baseline at each frame length.

2 Related Work

2.1 Video Diffusion Models

Building on denoising diffusion (Ho et al., 2020; Song et al., 2020), video generators have evolved from image backbones extended with temporal modules (Ho et al., 2022; Guo et al., 2024; Wang et al., 2024b) and latent video diffusion (Blattmann et al., 2023) to Diffusion Transformers (Peebles & Xie, 2023; Yang et al., 2025; Brooks et al., 2024; Kuaishou, 2024; Wan et al., 2025; Polyak et al., 2024; An et al., 2026a;b) that scale to large datasets and long videos (An et al., 2026c). These foundation models are primarily text- or image-conditioned and lack explicit camera control for video-to-video re-rendering, motivating the camera-controllable methods reviewed below.

2.2 Camera-Controlled Video Re-rendering

Camera-controlled generation injects camera poses or motion descriptors into a pretrained video generator (Wang et al., 2024c; He et al., 2025; Bahmani et al., 2025; Hou & Chen, 2025; Cheong et al., 2024; Luo et al., 2025; Ge et al., 2026). Early systems such as GCD (Van Hoorick et al., 2024) and ReCapture (Zhang et al., 2025a) establish the setting of re-rendering an existing monocular video from new viewpoints. Methods for this task differ mainly in how scene geometry reaches the generator. Latent-transfer methods (Bai et al., 2025; Jeong et al., 2025; Xie et al., 2026; Paliwal et al., 2026) condition on source-video tokens or learned scene latents together with the target camera and leave source-to-target correspondence to the network. Rendered-geometry methods (Zhang et al., 2025a; Ren et al., 2025; Yu et al., 2025; Chen et al., 2026; Cao et al., 2026; Qiao et al., 2025) reconstruct a scene proxy, typically from monocular depth, and render it into the target views as guidance; Trajectory Attention (Xiao et al., 2025b) conditions on pixel trajectories obtained by the same kind of reprojection. Such guidance is explicit but formed densely and frame by frame. Track2View keeps the correspondence explicit but conveys it through paired 3D point tracks instead of rendered images.

2.3 Point Tracks as Conditioning Signals

Point tracking has progressed from 2D trackers that follow many points jointly and through occlusion (Karaev et al., 2024; 2025) to 3D trackers that jointly model scene geometry, camera ego-motion, and object motion (Xiao et al., 2024b; 2025a), while feed-forward reconstruction methods recover point maps, cameras, and tracks without per-scene optimization (Wang et al., 2024a; Zhang et al., 2025b; Wang et al., 2025; Feng et al., 2025). Trajectories have accordingly become control signals for video generation, conditioning on 2D or 3D motion paths (Yin et al., 2023; Zhang et al., 2025c; Xiao et al., 2025b; 2024a). Motion Prompting (Geng et al., 2025) conditions on 2D track embeddings and obtains camera control by reprojecting estimated depth into 2D tracks. DaS (Gu et al., 2025) renders 3D tracks, colored by their coordinates, into a tracking video that conditions an image-to-video model; for re-rendering, its appearance comes from the first frame only, and the tracks serve as a rendered control video rather than as a channel that transports source content. Edit-by-Track (Lee et al., 2025) is closest to our setting: it conditions V2V generation on 3D point tracks for joint camera and object motion editing. Because its target tracks come from edited point clouds and are estimated independently of the source tracks, it relates the two views through learned cross-attention for sampling and splatting. Track2View addresses frame-synchronized re-rendering, where source and target tracks are the same 3D points by construction (Sec. 3.3). This lets fixed bilinear sampling and scattering replace the learned interface; in addition, placing 1,152 SpatialTrackerV2 queries at two frames improves Mat.Pix. and RotErr over 576 first-frame queries (Tab. 2).

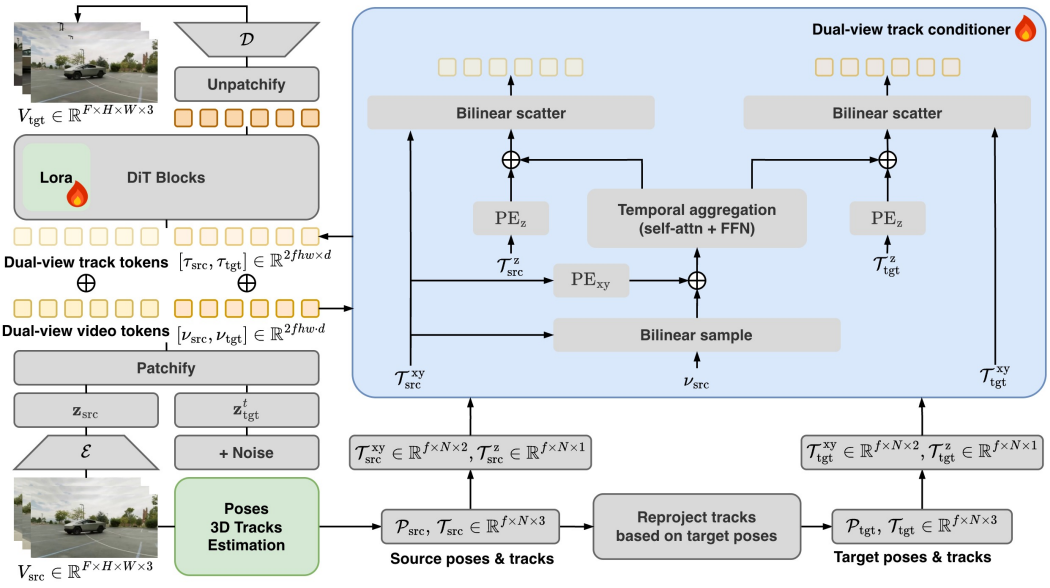

Figure 2: **Overview of Track2View.** $\oplus$ denotes element-wise addition; PE denotes Fourier positional encoding; N denotes the number of point tracks.

3 METHOD

The camera-controlled video re-rendering task can be defined as: given a source video $V_{\text{src}} \in \mathbb{R}^{F\times H\times W\times 3}$ and a user-specified target camera trajectory $\mathcal{P}_{\text{tgt}}$, the goal is to generate a frame-synchronized target video $V_{\text{tgt}} \in \mathbb{R}^{F\times H\times W\times 3}$ that faithfully reproduces the scene content and dynamics of V_{src} as observed from the target viewpoint, where each target frame V^t_{tgt} corresponds to the same timestep as V^t_{src}. To enable track-based conditioning, we use SpatialTrackerV2 (Xiao et al., 2025a) to jointly estimate camera poses $\mathcal{P}_{\text{src}}$ and 3D point tracks $\mathcal{T}_{\text{src}} \in \mathbb{R}^{F\times N\times 3}$ from the source video, and reproject $\mathcal{T}_{\text{src}}$ under $\mathcal{P}_{\text{tgt}}$ to obtain the corresponding target tracks $\mathcal{T}_{\text{tgt}}$.

3.1 FRAMEWORK

As illustrated in Fig. 2, the estimated tracks are projected into paired screen-space coordinates $(\mathcal{T}^{xy}_{\text{src}}, \mathcal{T}^{xy}_{\text{tgt}}) \in \mathbb{R}^{f\times N\times 2}$ and per-view depths $(\mathcal{T}^{z}_{\text{src}}, \mathcal{T}^{z}_{\text{tgt}}) \in \mathbb{R}^{f\times N\times 1}$.

The source video is encoded by a 3D variational autoencoder (VAE) encoder $\mathcal{E}$ into a latent $\mathbf{z}_{\text{src}} = \mathcal{E}(V_{\text{src}})$. The 3D VAE temporally compresses the video by a factor of 4, yielding $f{=}21$ latent frames from $F{=}81$ input frames; point tracks are subsampled at the same stride to align with the latent frame rate. The latent is patchified into source tokens ν_{src}. A noisy target latent $\mathbf{z}^t_{\text{tgt}}$ is sampled by adding noise at diffusion timestep t, and patchified into target tokens ν_{tgt}. The two are concatenated along the frame dimension to form dual-view video tokens $[\nu_{\text{src}}, \nu_{\text{tgt}}] \in \mathbb{R}^{2fhw\times d}$. Our dual-view track conditioner (Sec. 3.2) then produces track tokens $[\tau_{\text{src}}, \tau_{\text{tgt}}] \in \mathbb{R}^{2fhw\times d}$, which are added element-wise to the video tokens before they are processed by the DiT blocks. We fine-tune the pretrained DiT with Low-Rank Adaptation (LoRA) adapters while keeping all other parameters frozen, allowing the model to learn track-conditioned generation without catastrophic forgetting of the base video prior. The DiT denoises the target portion of the concatenated tokens, producing a clean latent $\mathbf{z}^0_{\text{tgt}}$ that is unpatchified and decoded by the 3D VAE decoder $\mathcal{D}$ to yield the final output video $V_{\text{tgt}} = \mathcal{D}(\mathbf{z}^0_{\text{tgt}})$.

3.2 DUAL-VIEW TRACK CONDITIONER

The dual-view track conditioner converts sparse paired 3D tracks into dense conditioning tokens that inject geometric correspondence into the diffusion process. All geometric operations—sampling and

scattering—use deterministic bilinear interpolation with no learnable parameters, ensuring that camera geometry is encoded directly rather than memorized; only the temporal aggregation and feature projections are learned, keeping the module lightweight and generalizable to unseen trajectories.

Given N 3D scene points tracked across f frames, we project them into both the source and target camera views using their respective camera parameters, obtaining normalized 2D coordinates $\mathcal{T}^{xy} \in \mathbb{R}^{f\times N\times 2}$ and camera-space depths $\mathcal{T}^{z} \in \mathbb{R}^{f\times N\times 1}$ for each view, along with a validity mask $\mathbf{m} \in \{0,1\}^{f\times N}$ marking tracks with positive depth that fall within the image bounds.

We extract per-track visual features by sampling from the source video tokens $\nu_{\text{src}} \in \mathbb{R}^{f\times hw\times d}$ at the projected source coordinates $\mathcal{T}^{xy}_{\text{src}}$ via bilinear grid sampling, yielding sparse track features $\hat{\tau} \in \mathbb{R}^{f\times N\times d}$. We then add a Fourier positional encoding of the sampling coordinates and apply a residual MLP for refinement:

$$\hat{\tau} \leftarrow \hat{\tau} + \mathrm{PE}_{xy}(\mathcal{T}^{xy}_{\text{src}}) + \mathrm{MLP}(\hat{\tau} + \mathrm{PE}_{xy}(\mathcal{T}^{xy}_{\text{src}})). \tag{1}$$

Invalid tracks (outside the source view) are zeroed out.

Each 3D point is observed across multiple frames, providing complementary visual information from different timesteps. We aggregate this per track across the temporal dimension using an L-layer transformer encoder (L=8 by default) with a padding mask that excludes frames where a track is occluded or out of view:

$$\bar{\tau} = \mathrm{TemporalAgg}(\hat{\tau}), \quad \bar{\tau} \in \mathbb{R}^{f\times N\times d}. \tag{2}$$

This propagates visual context from frames where a point is visible to frames where it is not, establishing temporal correspondences without relying on the DiT's own attention.

After temporal aggregation, we inject view-specific 3D structure by encoding the inverse depth $1/\mathcal{T}^{z}$ (disparity), jointly normalized across the source and target views to preserve their relative scale, via Fourier features:

$$v_{\text{src}} = \bar{\tau} + \mathrm{PE}_{z}(\mathcal{T}^{z}_{\text{src}}), \quad v_{\text{tgt}} = \bar{\tau} + \mathrm{PE}_{z}(\mathcal{T}^{z}_{\text{tgt}}). \tag{3}$$

The temporal features $\bar{\tau}$ are shared between both views since they describe the same 3D points, while the depth encodings differentiate the two viewpoints, allowing the model to reason about parallax and occlusion.

Finally, we scatter the per-track features back to dense spatial grids using the inverse of bilinear interpolation, where each track distributes its feature to the four nearest grid cells with bilinear weights, and cells receiving multiple contributions are averaged:

$$\tau_{\text{src}} = \mathrm{Scatter}(v_{\text{src}},\ \mathcal{T}^{xy}_{\text{src}}), \quad \tau_{\text{tgt}} = \mathrm{Scatter}(v_{\text{tgt}},\ \mathcal{T}^{xy}_{\text{tgt}}). \tag{4}$$

The resulting dual-view track tokens $[\tau_{\text{src}}, \tau_{\text{tgt}}] \in \mathbb{R}^{2fhw\times d}$ are added element-wise to the video tokens before the DiT blocks. Because both views share the same underlying 3D points, this mechanism naturally enforces consistent rendering across viewpoints.

During training, we apply source frame dropout (randomly zeroing up to 50% of source frames) and track subsampling (retaining 50–100% of tracks) to improve robustness to varying input conditions.

3.3 Training Data Curation

Our training requires paired videos of the same dynamic scene captured from two different camera viewpoints, together with one-to-one 3D point track correspondences between them. We source our data from the MultiCamVideo dataset of ReCamMaster (Bai et al., 2025), which consists of synthetic scenes rendered in Unreal Engine 5 where all virtual cameras are initialized at the same position, ensuring that they share an identical first frame. For each scene, we randomly select two cameras A and B to form a training pair, treating camera A as the source and camera B as the target, where each video contains F=81 frames.

To obtain one-to-one point track correspondences between the source and target videos, we temporally reverse camera A and concatenate it with camera B, yielding a single $2F-1$=161-frame sequence. Because all cameras share the same first frame, the last frame of the reversed camera A (i.e., frame 0 of the original) is identical to the first frame of camera B, producing a seamless transition at the concatenation boundary. We run SpatialTrackerV2 (Xiao et al., 2025a) on the concatenated sequence,



Figure 3: **Paired track extraction.** We temporally reverse the source video (camera A) and concatenate it with the target video (camera B), forming a 161-frame sequence with a shared first frame. Top: source frames (reversed back to original order). Bottom: target frames. Colored dots denote 3D points tracked by SpatialTrackerV2 (Xiao et al., 2025a) with queries at frames 0 and 80; matching colors across rows indicate one-to-one correspondences.

initializing query points at frames 0 and 80 only—we do not place queries beyond frame 80 because at inference time only the source video (81 frames) is available. Since the tracker follows each point continuously through the entire sequence, the resulting tracks naturally span both the source and target segments, establishing one-to-one correspondences without any explicit matching. The tracks from the first 81 frames (reversed camera A) are re-reversed to recover the original temporal order, giving us paired track sets $(\mathcal{T}_{\text{src}}, \mathcal{T}_{\text{tgt}}) \in \mathbb{R}^{2\times F\times N\times 3}$ ready for training. As shown in Fig. 3, the extracted tracks maintain consistent point identities across the source and target views. At inference time, the target tracks are obtained by reprojecting the source 3D point tracks under the user-specified target camera poses, requiring no additional tracking. We ablate the effect of query frame selection in Tab. 2.

3.4 Training Details

We build Track2View on Wan-2.1 (Wan et al., 2025) (Wan2.1-T2V-1.3B), a pretrained text-to-video diffusion transformer that generates 81-frame videos at 832×480 resolution. The dual-view track conditioner is trained from scratch with full parameter updates, while the pretrained DiT blocks are adapted using LoRA (Hu et al., 2022) with rank r=64 and α=64, applied to the query, key, value, output projection, and feed-forward layers of each transformer block. All other parameters of the base model remain frozen.

By default we set N=1152, with half of the queries placed at the first frame and half at the middle frame (t=80) of the temporally concatenated $2F-1$ sequence used for paired-track extraction (Sec. 3.3). During training, the source latent $\mathbf{z}_{\text{src}}$ remains clean (no noise added), while noise is applied only to the target latent $\mathbf{z}_{\text{tgt}}$. The training objective is the standard flow-matching loss (Lipman et al., 2023) computed on the target portion of the concatenated sequence only, ensuring the model learns to denoise the target view conditioned on the clean source. We train with AdamW at a learning rate of 1×10^{-5} in bfloat16 precision using DeepSpeed ZeRO Stage 1. We train on 4 A100 GPUs with a batch size of 2 per GPU for 216K optimization steps.

At inference time, we use 50 denoising steps with classifier-free guidance scale 2.0 applied to the text prompt.

4 EXPERIMENTS

4.1 DATASET AND EVALUATION PROTOCOL

We evaluate on the RealCam-Vid benchmark (Zheng et al., 2025). Our evaluation set comprises 400 videos: 200 static scenes sourced from RealEstate10K (Zhou et al., 2018) and 200 dynamic scenes from MiraData (Ju et al., 2024), enabling separate assessment under different scene conditions. FID reference features are pre-extracted from 33,388 real frames (Inception-v3) sampled from the RealCam-Vid test set. We use the text captions provided by the benchmark as the text conditions.

4.2 METRICS

We evaluate along three complementary axes.

Visual quality is measured by FID↓ (Fréchet Inception Distance), which computes the Fréchet distance between Inception-v3 features of generated frames and those of the real video pool; CLIP-T↑, which measures text-video semantic alignment via CLIP ViT-B/32 between generated frames and the text prompt; and CLIP-F↑, which measures temporal consistency as the average CLIP cosine similarity between consecutive generated frames.

View synchronization is measured by CLIP-V↑, the average CLIP cosine similarity between source and generated frames at the same timestamp; and Mat.Pix. (K)↑, which counts confidently matched keypoints (in thousands) between source and generated frame pairs using GIM-LightGlue (Shen et al., 2024), providing a geometry-aware correspondence signal.

Camera accuracy is measured by RotErr↓ and TransErr↓, following the protocol of CameraCtrl (He et al., 2025). We extract camera poses from generated videos using GLOMAP (Pan et al., 2024) (via COLMAP (Schönberger & Frahm, 2016) global mapper). RotErr reports the mean geodesic distance (in degrees) between ground-truth and estimated rotation matrices, and TransErr reports the mean ℓ_2 distance between translation vectors after scale alignment to the first two frames. Both metrics compare extracted poses against the ground-truth target trajectory.

4.3 BASELINES

We compare Track2View against four state-of-the-art camera-controlled video generation methods: Trajectory Attention (Xiao et al., 2025b), TrajectoryCrafter (Yu et al., 2025), Gen3C (Ren et al., 2025), and ReCamMaster (Bai et al., 2025), as well as the track-conditioned generator DaS (Gu et al., 2025). Since Trajectory Attention and TrajectoryCrafter parameterize camera motion relative to the source frame, we evaluate them in a fair V2V mode, supplying per-frame camera conditions derived from the same source video and target trajectory as Track2View. DaS is built on an image-to-video backbone, so we run it in I2V mode with its default setting at its native 49-frame length (denoted * in Tab. 1).

4.4 QUANTITATIVE COMPARISONS

Quantitative results are reported in Tab. 1. Since Trajectory Attention generates 25 frames and TrajectoryCrafter and DaS generate 49 frames, we also evaluate Track2View at matching frame counts by taking the first 25 or 49 frames of our 81-frame output. Track2View consistently outperforms all baselines across the three evaluation axes.

In visual quality, Track2View achieves the lowest FID at every frame length (e.g., 26.82 vs. 30.32 for Gen3C and 33.85 for ReCamMaster at 81 frames), while also achieving the highest CLIP-T and CLIP-F at every frame length, confirming that the gains do not come at the cost of semantic fidelity.

For view synchronization, Track2View achieves the highest CLIP-V (93.22 vs. 92.93 for ReCamMaster) and Mat.Pix. (0.695K vs. 0.644K for Gen3C and 0.579K for ReCamMaster) at 81 frames, confirming stronger geometric correspondence between source and generated views. Gains are most pronounced at 25 frames, where Mat.Pix. reaches 1.070K compared to 0.963K for Trajectory Attention (+11.1%).

Table 1: Quantitative comparison on the RealCam-Vid benchmark (400 videos). Trajectory Attention and TrajectoryCrafter are evaluated in fair V2V mode, receiving per-frame camera conditions derived from the same source video and target trajectory as Track2View. *Evaluated in I2V mode.

		Visual Quality			*View Sync.*		*Cam. Acc.*	
Method	Frames	FID↓	CLIP-T↑	CLIP-F↑	CLIP-V↑	Mat.Pix.(K)↑	RotErr(°)↓	TransErr↓
Trajectory Attention (Xiao et al., 2025b)	25	40.72	28.63	98.84	94.37	0.963	1.55	0.203
Track2View (Ours)	25	**33.95**	**29.06**	**99.28**	**95.81**	**1.070**	**1.24**	**0.085**
TrajectoryCrafter (Yu et al., 2025)	49	33.40	28.80	98.99	93.59	0.796	2.14	0.815
DaS* (Gu et al., 2025)	49	29.95	28.39	98.46	91.88	0.682	3.59	3.88
Track2View (Ours)	49	**29.30**	**29.07**	**99.35**	**94.71**	**0.876**	**1.31**	**0.280**
Gen3C (Ren et al., 2025)	81	30.32	28.55	99.04	92.47	0.644	4.21	3.715
ReCamMaster (Bai et al., 2025)	81	33.85	28.48	99.33	92.93	0.579	2.20	2.096
Track2View (Ours)	81	**26.82**	**28.89**	**99.38**	**93.22**	**0.695**	**1.55**	**0.818**

Table 2: Ablation on query strategy.

Configuration	CLIP-V↑	Mat.Pix.↑	RotErr↓	TransErr↓
First only (576)	92.79	0.682	2.36	**1.574**
Rand. 1/2 (576)	92.79	0.681	**2.24**	1.797
Rand. 1/4 (288)	**92.80**	0.680	2.36	2.440
All (1152)	**92.80**	**0.685**	**2.24**	<u>1.681</u>

Table 3: Ablation on temporal aggregation depth.

Configuration	CLIP-V↑	Mat.Pix.↑	RotErr↓	TransErr↓
2 layers	92.80	0.685	2.24	1.681
4 layers	92.52	0.679	1.97	1.214
8 layers	**93.22**	**0.695**	**1.55**	**0.818**

Camera accuracy sees the most significant improvements: relative to the strongest baseline at each frame length, RotErr drops by 20–39% and TransErr by 58–66%. The largest relative gains appear at 49 frames, where Track2View reduces RotErr from 2.14° to 1.31° (-39%) and TransErr from 0.815 to 0.280 (-66%) compared to TrajectoryCrafter. Absolute errors naturally grow at 81 frames as pose estimation errors accumulate over longer sequences; nonetheless, Track2View still reduces RotErr by 30% and TransErr by 61% compared to the strongest baseline at this length.

Against the track-conditioned DaS at 49 frames, Track2View improves CLIP-V (94.71 vs. 91.88) and Mat.Pix. (0.876K vs. 0.682K), and reduces RotErr from 3.59° to 1.31° and TransErr from 3.88 to 0.280. This indicates that transporting source-video features along paired tracks yields substantially tighter camera control and view synchronization than conditioning an I2V model on a rendered tracking video.

On six VBench (Huang et al., 2024) dimensions (Appendix B), Track2View achieves the best or second-best score on every dimension at every frame length.

4.5 Qualitative Comparisons

Fig. 4 presents qualitative comparisons with Gen3C (Ren et al., 2025) and ReCamMaster (Bai et al., 2025) on two representative scenes. In the dynamic scene (top), Gen3C introduces floating artifacts and inconsistent foreground objects and background geometry due to noisy per-frame depth warping, while ReCamMaster fails to follow the prescribed camera trajectory and hallucinates objects absent from the source video (e.g., a red scooter in the background). In the static scene (bottom), both baselines exhibit progressive content drift as the camera moves upward—new objects appear or disappear in later frames. Our method avoids these failure modes by conditioning on paired 3D tracks that anchor generated content to the source video's geometry, producing temporally stable outputs that preserve scene content faithfully even under large camera motions.

4.6 Ablation Studies

We ablate the track query strategy (Tab. 2) and the temporal aggregation depth (Tab. 3) on the full 400-video benchmark at 81 frames.

Tab. 2 compares query-frame strategies for SpatialTrackerV2 (Xiao et al., 2025a), all with 2 temporal aggregation layers. "All" (queries at the first and middle frames, 1152 tracks) gives the best Mat.Pix. and tied-best RotErr; "First only" attains slightly lower TransErr (1.574 vs. 1.681), but mid-sequence queries cover content revealed only after the first frame, which we find essential with the deeper

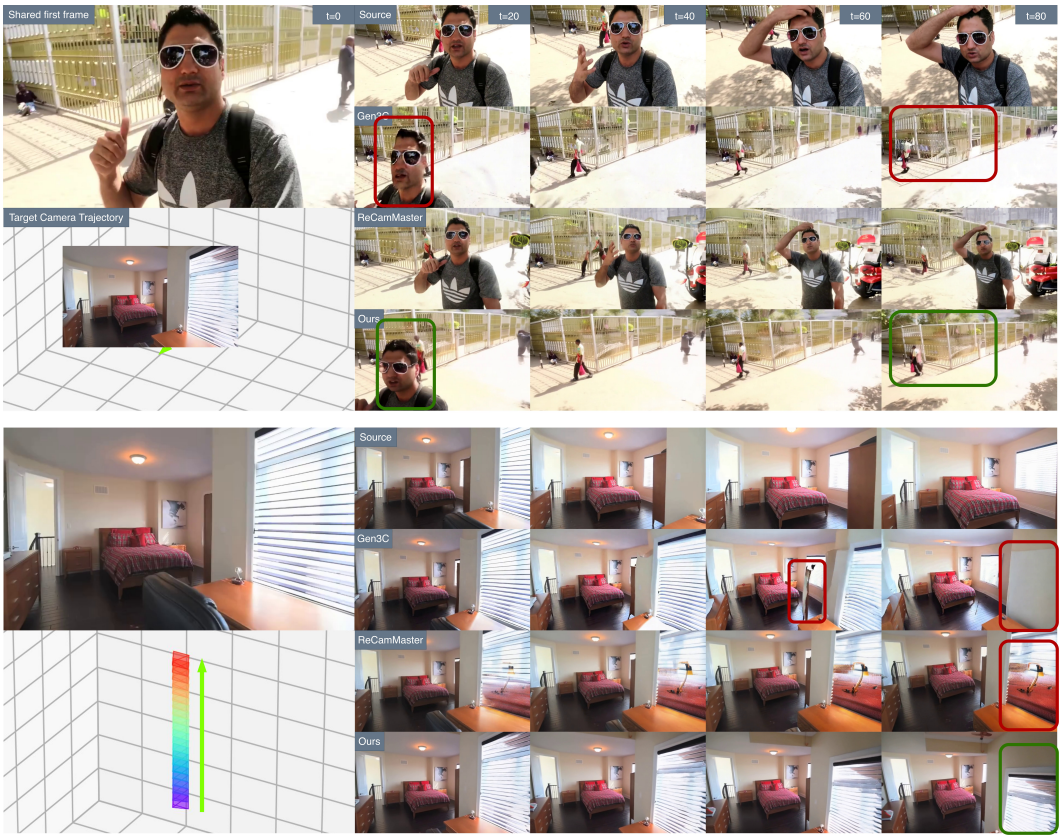

Figure 4: Qualitative comparison on a dynamic outdoor scene (top) and a static indoor scene (bottom). Red boxes highlight artifacts or hallucinated content; green boxes indicate faithful generation by our method.

aggregation of our final model. Subsampling to 288 tracks degrades TransErr to 2.440, showing that track density matters for camera control. We adopt "All" by default.

Tab. 3 varies the number of self-attention layers in the temporal aggregation module. Camera accuracy improves monotonically with depth: RotErr drops from $2.24°$ to $1.55°$ (-31%) and TransErr from 1.681 to 0.818 (-51%) as we go from 2 to 8 layers. Overall, deeper temporal aggregation more effectively propagates visual context across frames, and we use 8 layers as our default.

5 Conclusion

We presented Track2View, a camera-controlled video re-rendering framework that conditions a video diffusion transformer on paired 3D point tracks. By representing camera motion as sparse trajectories projected into both source and target screen spaces, our method provides explicit, temporally continuous correspondences that existing pose-based and rendering-based approaches lack. The dual-view track conditioner uses parameter-free bilinear sampling and scattering for geometry encoding, combined with learned temporal aggregation for cross-frame context propagation. Experiments on a 400-video benchmark spanning static and dynamic scenes demonstrate that Track2View achieves state-of-the-art results across visual quality, view synchronization, and camera accuracy, reducing rotation and translation errors by 20–39% and 58–66%, respectively compared to leading baselines.

AI use statement

We used generative AI tools for writing only: to draft sections of this paper and to aid and polish the writing throughout. Every AI-drafted passage was read and edited by the authors, and every

passage making a technical claim was verified against our source code and evaluation outputs before inclusion.

We did not use generative AI tools to generate or retrieve any reference.

We have reviewed all AI-assisted work and take responsibility for the final content of this paper, including text, claims and artifacts produced with the aid of generative AI.

REFERENCES

Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

Zhaochong An, Menglin Jia, Haonan Qiu, Zijian Zhou, Xiaoke Huang, Zhiheng Liu, Weiming Ren, Kumara Kahatapitiya, Ding Liu, Sen He, et al. Onestory: Coherent multi-shot video generation with adaptive memory. *CVPR*, 2026a.

Zhaochong An, Orest Kupyn, Théo Uscidda, Andrea Colaco, Karan Ahuja, Serge Belongie, Mar Gonzalez-Franco, and Marta Tintore Gazulla. Vggrpo: Towards world-consistent video generation with 4d latent reward. *arXiv preprint arXiv:2603.26599*, 2026b.

Zhaochong An, Zirui Li, Mingqiao Ye, Feng Qiao, Jiaang Li, Zongwei Wu, Vishal Thengane, Chengzu Li, Lei Li, Luc Van Gool, et al. Video understanding: From geometry and semantics to unified models. *Machine Intelligence Research*, 2026c.

Sherwin Bahmani, Ivan Skorokhodov, Guocheng Qian, Aliaksandr Siarohin, Willi Menapace, Andrea Tagliasacchi, David B Lindell, and Sergey Tulyakov. Ac3d: Analyzing and improving 3d camera control in video diffusion transformers. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22875–22889, 2025.

Jianhong Bai, Menghan Xia, Xiao Fu, Xintao Wang, Lianrui Mu, Jinwen Cao, Zuozhu Liu, Haoji Hu, Xiang Bai, Pengfei Wan, et al. Recammaster: Camera-controlled generative rendering from a single video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14834–14844, 2025.

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

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Leo Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

Wei Cao, Hao Zhang, Fengrui Tian, Yulun Wu, Yingying Li, Shenlong Wang, Ning Yu, and Yaoyao Liu. Freeorbit4d: Training-free arbitrary camera redirection for monocular videos via geometry-complete 4d reconstruction. *arXiv preprint arXiv:2601.18993*, 2026.

Dong-Yu Chen, Yixin Guo, Shuojin Yang, Tai-Jiang Mu, and Shi-Min Hu. Beyond inpainting: Unleash 3d understanding for precise camera-controlled video generation. *arXiv preprint arXiv:2601.10214*, 2026.

Soon Yau Cheong, Duygu Ceylan, Armin Mustafa, Andrew Gilbert, and Chun-Hao Paul Huang. Boosting camera motion control for video diffusion transformers. *arXiv preprint arXiv:2410.10802*, 2024.

Haiwen Feng, Junyi Zhang, Qianqian Wang, Yufei Ye, Pengcheng Yu, Michael J Black, Trevor Darrell, and Angjoo Kanazawa. St4rtrack: Simultaneous 4d reconstruction and tracking in the world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8503–8513, 2025.

Wenhang Ge, Guibao Shen, Jiawei Feng, Luozhou Wang, Hao Lu, Xingye Tian, Xin Tao, and Ying-Cong Chen. Campilot: Improving camera control in video diffusion model with efficient camera reward feedback. *arXiv preprint arXiv:2601.16214*, 2026.

Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Carl Doersch, Yusuf Aytar, Michael Rubinstein, Chen Sun, Oliver Wang, Andrew Owens, and Deqing Sun. Motion prompting: Controlling video generation with motion trajectories. 2025.

Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, et al. Kubric: A scalable dataset generator. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3749–3761, 2022.

Zekai Gu, Rui Yan, Jiahao Lu, Peng Li, Zhiyang Dou, Chenyang Si, Zhen Dong, Qifeng Liu, Cheng Lin, Ziwei Liu, et al. Diffusion as shader: 3d-aware video diffusion for versatile video generation control. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pp. 1–12, 2025.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *International Conference on Learning Representations*, 2024.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for video diffusion models. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Chen Hou and Zhibo Chen. Training-free camera control for video generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Hyeonho Jeong, Suhyeon Lee, and Jong Chul Ye. Reangle-a-video: 4d video generation as video-to-video translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 11164–11175, 2025.

Xuan Ju, Yiming Gao, Zhaoyang Zhang, Ziyang Yuan, Xintao Wang, Ailing Zeng, Yu Xiong, Qiang Xu, and Ying Shan. Miradata: A large-scale video dataset with long durations and structured captions. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *European conference on computer vision*, pp. 18–35. Springer, 2024.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6013–6022, 2025.

Jay Karhade, Nikhil Keetha, Yuchen Zhang, Tanisha Gupta, Akash Sharma, Sebastian Scherer, and Deva Ramanan. Any4d: Unified feed-forward metric 4d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14578–14589, 2026.

Kuaishou. Kling video model. https://kling.kuaishou.com/en, 2024.

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. *arXiv preprint arXiv:2512.02015*, 2025.

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

Yawen Luo, Xiaoyu Shi, Jianhong Bai, Menghan Xia, Tianfan Xue, Xintao Wang, Pengfei Wan, Di Zhang, and Kun Gai. Camclonemaster: Enabling reference-based camera control for video generation. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–10, 2025.

Avinash Paliwal, Adithya Iyer, Shivin Yadav, Muhammad Ali Afridi, and Midhun Harikumar. Reshoot-anything: A self-supervised model for in-the-wild video reshooting. *arXiv preprint arXiv:2604.21776*, 2026.

Linfei Pan, Daniel Barath, Marc Pollefeys, and Johannes L. Schönberger. Global structure-from-motion revisited. In *European Conference on Computer Vision (ECCV)*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pp. 4195–4205, 2023.

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*, 2024.

Feng Qiao, Zhexiao Xiong, Eric Xing, and Nathan Jacobs. Towards open-world generation of stereo images and unsupervised matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6121–6132, 2025.

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

Xuelun Shen, Zhipeng Cai, Wei Yin, Matthias Müller, Zijun Li, Kaixuan Wang, Xiaozhi Chen, and Cheng Wang. Gim: Learning generalizable image matcher from internet videos. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

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

Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. In *European Conference on Computer Vision*, pp. 313–331. Springer, 2024.

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*, 2025.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 5294–5306, 2025.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20697–20709, 2024a.

Yaohui Wang, Xinyuan Chen, Xin Ma, Shangchen Zhou, Ziqi Huang, Yi Wang, Ceyuan Yang, Yinan He, Jiashuo Yu, Peiqing Yang, et al. Lavie: High-quality video generation with cascaded latent diffusion models. *IJCV*, 2024b.

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 2024 Conference Papers*, pp. 1–11, 2024c.

FU Xiao, Xian Liu, Xintao Wang, Sida Peng, Menghan Xia, Xiaoyu Shi, Ziyang Yuan, Pengfei Wan, Di Zhang, and Dahua Lin. 3dtrajmaster: Mastering 3d trajectory for multi-entity motion in video generation. In *The Thirteenth International Conference on Learning Representations*, 2024a.

Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20406–20417, 2024b.

Yuxi Xiao, Jianyuan Wang, Nan Xue, Nikita Karaev, Yuri Makarov, Bingyi Kang, Xing Zhu, Hujun Bao, Yujun Shen, and Xiaowei Zhou. Spatialtrackerv2: Advancing 3d point tracking with explicit camera motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6726–6737, 2025a.

Zeqi Xiao, Wenqi Ouyang, Yifan Zhou, Shuai Yang, Lei Yang, Jianlou Si, and Xingang Pan. Trajectory attention for fine-grained video motion control. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Mingyang Xie, Numair Khan, Tianfu Wang, Naina Dhingra, Seonghyeon Nam, Haitao Yang, Zhuo Hui, Christopher Metzler, Andrea Vedaldi, Hamed Pirsiavash, et al. Lavr: Scene latent conditioned generative video trajectory re-rendering using large 4d reconstruction models. *arXiv preprint arXiv:2601.14674*, 2026.

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

Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. Dragnuwa: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv preprint arXiv:2308.08089*, 2023.

Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 100–111, 2025.

David Junhao Zhang, Roni Paiss, Shiran Zada, Nikhil Karnad, David E Jacobs, Yael Pritch, Inbar Mosseri, Mike Zheng Shou, Neal Wadhwa, and Nataniel Ruiz. Recapture: Generative video camera controls for user-provided videos using masked video fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2050–2062, 2025a.

Junyi Zhang, Charles Herrmann, Junhwa Hur, Varun Jampani, Trevor Darrell, Forrester Cole, Deqing Sun, and Ming-Hsuan Yang. Monst3r: A simple approach for estimating geometry in the presence of motion. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Zhenghao Zhang, Junchao Liao, Menghao Li, Zuozhuo Dai, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. Tora: Trajectory-oriented diffusion transformer for video generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 2063–2073, 2025c.

Guangcong Zheng, Teng Li, Xianpan Zhou, and Xi Li. Realcam-vid: High-resolution video dataset with dynamic scenes and metric-scale camera movements. *arXiv preprint arXiv:2504.08212*, 2025.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *ACM SIGGRAPH*, 2018.

A Implementation and Evaluation Details

Inference. Track2View samples with 50 denoising steps and a classifier-free guidance scale of 2.0 on the text prompt.

Baselines on RealCam-Vid. Trajectory Attention (Xiao et al., 2025b) and TrajectoryCrafter (Yu et al., 2025) define the target camera relative to a static source camera. In our fair V2V protocol, we give both methods the per-frame relative transform $T_{\text{src}\to\text{tgt}}(i) = \mathbf{w2c}_{\text{tgt}}(i)\,\mathbf{c2w}_{\text{src}}(i)$, computed from the SpatialTrackerV2 source poses and the benchmark target trajectory, so that both the moving source camera and the target camera are respected. For TrajectoryCrafter, we replace its static-source pose construction with these transforms and align its DepthCrafter depth to the scale of our tracks (median relative depth error 0.7%). For Trajectory Attention, we use the same per-frame transforms, align its monocular depth to the track scale with an affine fit, and use CoTracker (Karaev et al., 2024) for source object motion. Tab. 4 compares this protocol with the methods' native I2V interface: the fair protocol substantially improves the camera accuracy of Trajectory Attention (RotErr $3.54^\circ \to 1.55^\circ$), which confirms that the supplied poses are informative, but both baselines remain behind Track2View on every metric in Tab. 1. DaS (Gu et al., 2025) is run with its released pipeline and default settings (50 steps, guidance 6.0, 4,900 tracks from its own tracker on a 70×70 grid); only the target camera trajectory comes from the benchmark, rescaled to the DaS depth scale by the median depth ratio. DaS generates 49 frames at 720×480, which we resize to the evaluation resolution.

Table 4: Native I2V versus fair V2V protocol for the two relative-camera baselines on RealCam-Vid (400 videos). Track2View results at the same frame counts are in Tab. 1.

Method	Protocol	Frames	FID↓	CLIP-T↑	CLIP-F↑	CLIP-V↑	Mat.Pix.(K)↑	RotErr(°)↓	TransErr↓
Trajectory Attention	native I2V	25	38.25	28.57	98.49	94.87	0.826	3.54	0.309
Trajectory Attention	fair V2V	25	40.72	28.63	98.84	94.37	0.963	1.55	0.203
TrajectoryCrafter	native I2V	49	29.34	28.87	98.70	94.70	0.618	2.12	0.72
TrajectoryCrafter	fair V2V	49	33.40	28.80	98.99	93.59	0.796	2.14	0.815

Camera metrics. RotErr is averaged over videos that GLOMAP registers. Monocular SfM occasionally diverges in scale, producing translation errors several orders of magnitude above the rest; such videos are excluded from the TransErr mean.

B VBench Evaluation

We report six VBench (Huang et al., 2024) dimensions—Aesthetic Quality, Imaging Quality, Temporal Flickering, Motion Smoothness, Subject Consistency, and Background Consistency—to assess perceptual and temporal quality (Tab. 5).

Table 5: VBench (Huang et al., 2024) evaluation. Trajectory Attention and TrajectoryCrafter are evaluated in fair V2V mode; DaS is evaluated in I2V mode.

Method	Frames	Aesthetic Quality↑	Imaging Quality↑	Temporal Flickering↑	Motion Smoothness↑	Subject Consistency↑	Background Consistency↑
Trajectory Attention (Xiao et al., 2025b)	25	50.31	67.32	96.75	98.93	96.91	95.49
Track2View (Ours)	25	**53.54**	**70.28**	**96.94**	**99.26**	**97.66**	**95.72**
TrajectoryCrafter (Yu et al., 2025)	49	**54.04**	**70.71**	96.34	98.88	95.46	94.55
DaS (Gu et al., 2025)	49	51.02	68.20	96.54	98.84	92.51	93.90
Track2View (Ours)	49	53.87	70.11	**97.02**	**99.28**	**96.67**	**94.92**
Gen3C (Ren et al., 2025)	81	51.53	68.83	96.49	99.12	93.75	92.36
ReCamMaster (Bai et al., 2025)	81	52.29	67.52	**97.23**	**99.29**	95.26	93.47
Track2View (Ours)	81	**53.68**	**69.85**	97.13	**99.29**	**95.33**	**94.22**

C Kubric-4D Evaluation

Kubric-4D (Greff et al., 2022; Van Hoorick et al., 2024) provides synthetic dynamic scenes with ground-truth geometry, object motion, and cameras. It removes test-time estimation ambiguity, enabling comparisons in which every method receives exact geometry, and controlled ablations of the conditioning interface (App. D).

Moving-20 benchmark. The standard GCD protocol (Van Hoorick et al., 2024) keeps the source camera static. Since our task targets a moving source camera, we build *Moving-20* from the 20 test scenes: the source and target cameras share their first pose and then follow independent trajectories. Computed from input metadata only, the median maximum source–target baseline is 13.4 m (max. 33.3 m) and the median relative rotation is 57.7° (max. 130.9°); every scene contains an object that rotates by at least 90°, and 14 of 20 contain an object whose heading turns by at least 45°. On average, 30.6% of source track-time is invisible (occluded or out of view), and the median per-scene maximum drop in source-to-target warp coverage is 53.3 percentage points. Following GCD, source and ground-truth target videos are rendered from the merged 16-view point clouds at 420×280 and upsampled to 832×480.

Metrics. Following GCD, each scene is scored at 14 timesteps (frame index τs, $\tau = 0, \ldots, 13$, with a scene-specific stride $s \in \{1, \ldots, 4\}$) at 384×256 with PSNR, SSIM, and LPIPS (AlexNet). Within a frame-length block, only the timesteps inside that block's horizon are scored, so numbers are comparable within a block but not across blocks.

Kubric-4D training. All Kubric-4D models, including the ablation variants in App. D, are trained on the Kubric-4D training split: 156 scenes (disjoint from the test scenes) with 14 source/target pairs each (4 with a static source camera and 10 with a moving one), giving 2,184 pairs of 81 frames at 832×480. Each pair carries 24,960 ground-truth paired tracks (12,480 queried at frame 0 and 12,480 at frame 80). We train for 4 epochs on 2 GPUs with batch size 1 per GPU, learning rate 1×10^{-5}, and the LoRA configuration of the main model.

Tab. 6 compares Track2View with baselines that use their released weights. Every method receives ground-truth cameras, and geometry-based baselines receive ground-truth geometry wherever their interface accepts it: Gen3C uses ground-truth source depth; DaS uses a tracking video rendered from ground-truth tracks together with the first frame; Trajectory Attention and TrajectoryCrafter use the fair V2V protocol of App. A with ground-truth cameras. Track2View uses 12,480 ground-truth paired tracks. Our model obtains the best PSNR and SSIM in every block and the best LPIPS at 25 and 49 frames. At 81 frames, Gen3C, which receives ground-truth source depth, is close in PSNR (21.88 vs. 22.16 dB) and has lower LPIPS (0.187 vs. 0.194).

Table 6: Kubric-4D Moving-20 Evaluation. * denotes I2V.

Method	Frames	Setting	PSNR↑	SSIM↑	LPIPS↓
Trajectory Attention (Xiao et al., 2025b)	25	fair V2V	21.57	0.6760	0.2659
Track2View	25	12,480 GT paired tracks	**23.39**	**0.7121**	**0.1658**
TrajectoryCrafter (Yu et al., 2025)	49	fair V2V	18.65	0.5195	0.3343
DaS* (Gu et al., 2025)	49	GT-track rendering + first frame	18.43	0.4941	0.4067
Track2View	49	12,480 GT paired tracks	**22.23**	**0.6812**	**0.1917**
Gen3C (Ren et al., 2025)	81	GT source depth	21.88	0.6758	**0.1869**
ReCamMaster (Bai et al., 2025)	81	target cameras	16.76	0.4738	0.4764
Track2View	81	12,480 GT paired tracks	**22.16**	**0.6786**	0.1935

D Controlled Conditioning Ablations

To isolate the conditioning design, we train four variants that share everything except the conditioner: initialization, Kubric-4D training data, schedule, seed, number of GPUs, and Moving-20 evaluation (App. C). Results are in Tab. 7 and Tab. 8.

- **K1 (Track2View):** parameter-free bilinear sampling and scattering along paired tracks with the 8-layer temporal Transformer.
- **K2 (warped RGB):** the source video is forward-warped into the target view with *ground-truth* depth, VAE-encoded, concatenated with a latent validity mask, and linearly projected into the video tokens in place of the track tokens. No tracks are used, so track density does not apply.

- **K3 (mean aggregation):** identical to K1, except that the temporal Transformer is replaced by a masked mean over the frames in which each track is valid, followed by a learnable LayerNorm.
- **K4 (Edit-by-Track interface):** our reimplementation of the Edit-by-Track conditioner (Lee et al., 2025), whose code is not public. Tracks are represented by screen coordinates and normalized disparity; one coordinate cross-attention layer reads source tokens and one writes to the target grid; like the original, we omit the attention bias. The temporal aggregator is the same 8-layer Transformer as in K1, so K1 and K4 differ only in the read/write interface. We train and evaluate K4 at 1,000 tracks, the operating point of Edit-by-Track, and compare it with K1 at the same density.

Paired tracks vs. warped RGB (K1 vs. K2). Although K2 is given exact depth, K1 is better by 3.02 dB PSNR, 0.119 SSIM, and 0.114 LPIPS. A warped image is a frame-local projection: large relative rotations expose unseen regions (a median 53 pp coverage drop on Moving-20), and occlusion and reappearance break correspondence between frames. Paired tracks keep each point's identity and history across these gaps, and the conditioner transports source latent features instead of warped pixels.

Learned vs. mean temporal aggregation (K1 vs. K3). At every matched track density, the learned temporal Transformer outperforms the masked mean by 0.66–0.77 dB PSNR and 0.038–0.048 SSIM, and lowers LPIPS by 0.017–0.020 (Tab. 8). Together with the aggregation-layer ablation in Tab. 3, this shows that learned aggregation contributes beyond simple averaging.

Parameter-free transfer vs. Edit-by-Track (K1 vs. K4). At 1,000 tracks, K1 outperforms our Edit-by-Track reimplementation by 5.53 dB PSNR. The interfaces also differ in cost (Tab. 9): bilinear transfer is $O(N)$ in the number of tracks, whereas the cross-attention write is $O(Nhw)$. Our conditioner has 130.8 M parameters versus 248.0 M for the reimplementation. With 24,960 tracks, our Kubric-4D training density, the Edit-by-Track conditioner no longer fits in 80 GB during training, while ours needs 36.7 GB.

Table 7: Controlled conditioning ablation on Kubric-4D Moving-20 (81 frames, ground-truth geometry). All variants share initialization, training data, schedule, and evaluation; only the conditioner changes.

Variant	Source-to-target interface	Temporal agg.	Tracks	PSNR↑	SSIM↑	LPIPS↓
K1: Track2View	paired tracks, bilinear	Transformer	12,480	**22.16**	**0.6786**	**0.1935**
K2: Warped RGB	GT-depth warp + validity mask	–	dense	19.14	0.5598	0.3075
K3: Mean aggregation	paired tracks, bilinear	masked mean	12,480	21.39	0.6308	0.2130
K1: Track2View	paired tracks, bilinear	Transformer	1,000	**21.96**	**0.6716**	**0.2090**
K4: Edit-by-Track	paired tracks, cross-attention	Transformer	1,000	16.43	0.4403	0.4525

Table 8: Temporal aggregation across ground-truth track densities on Kubric-4D Moving-20 (81 frames). Frame-0 and frame-80 queries are balanced, and smaller sets are nested subsets of larger ones.

	K1: Transformer (Ours)			K3: masked mean		
Tracks	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
1,000	21.96	0.6716	0.2090	–	–	–
2,496	22.18	0.6745	0.1977	21.52	0.6367	0.2146
6,240	22.07	0.6774	0.1963	21.41	0.6323	0.2140
12,480	22.16	0.6786	0.1935	21.39	0.6308	0.2130
24,960	22.04	0.6748	0.1962	21.34	0.6311	0.2141

Depth encoding. We also ablate the view-specific disparity encoding PE_z (Sec. 3.2). Starting from K1, we continue training for one epoch (1,092 steps) with and without PE_z on the same hardware allocation and seed, and evaluate with 2,496 ground-truth tracks (Tab. 10). Removing PE_z costs 0.57 dB PSNR and increases LPIPS by 0.0079 after matched training; removing it only at inference from the model trained with it costs 0.61 dB. Per-view depth therefore carries information that the

Table 9: Peak GPU memory (GB) of the conditioner alone versus the number of tracks (H100 80 GB; latent grid $21 \times 60 \times 104$; d=1536; both use the same 8-layer temporal aggregator). Our Edit-by-Track reimplementation uses FlashAttention-backed scaled dot-product attention; a naive implementation would need more memory. OOM: out of memory.

	Inference		Training	
Tracks	Ours	Edit-by-Track	Ours	Edit-by-Track
1,000	5.94	7.83	10.72	20.38
4,000	8.10	12.28	12.13	32.00
12,480	18.73	25.61	22.72	65.63
24,960	30.48	45.22	36.70	OOM
49,920	45.90	OOM	OOM	OOM
Parameters	130.8 M	248.0 M		

2D track positions alone do not provide, such as the ordering of near and far surfaces that project to similar image locations.

Table 10: Ablation of the disparity encoding PE_z on Kubric-4D Moving-20 (81 frames, 2,496 ground-truth tracks). All rows continue training K1 for one additional epoch (5 epochs in total) with the same seed and hardware allocation.

Training	Inference	PSNR↑	SSIM↑	LPIPS↓
with PE_z	with PE_z	**22.60**	**0.6867**	**0.1892**
with PE_z	without PE_z	21.99	0.6786	0.2100
without PE_z	without PE_z	22.03	0.6829	0.1971

E Sparse versus Dense Tracks

Motion Prompting (Geng et al., 2025) finds dense tracks more effective than sparse ones when training on 2D tracks. We examine this question in our setting, where 3D tracks are *estimated*, at two levels: tracking accuracy and generation quality.

Tracking accuracy. We run Any4D (Karhade et al., 2026) on the 20 Moving-20 videos with a full 420×280 query grid at both frame 0 and frame 80 (235,200 candidate tracks per video) and compare subsets against the ground-truth tracks (Tab. 11). The full dense pool is less accurate than even a uniform subset in EPE, and a confidence-selected subset of 1,152 tracks (576 per query frame) improves all four metrics over the dense pool. Across the 40 scene/query-frame pairs, Any4D confidence is negatively correlated with per-track 3D error (mean Spearman $\rho = -0.589$). More tracks therefore do not imply more reliable geometry: a dense condition also passes on its least reliable correspondences.

Table 11: Any4D track accuracy on Kubric-4D Moving-20 against ground-truth tracks. All subsets are drawn from the same dense candidate pool. The oracle row selects the tracks with the lowest ground-truth error and is an upper bound.

Selection	Tracks	EPE (px)↓	Rel. 3D err.↓	Depth AbsRel↓	< 5 px↑
Dense (all query pixels)	235,200	9.344	0.1266	0.1034	65.18%
Uniform sparse	1,152	8.641	0.1293	0.1049	64.45%
Confidence top	1,152	**6.638**	**0.0595**	**0.0508**	**72.58%**
Oracle selection (upper bound)	1,152	2.504	0.0158	0.0107	86.93%

Generation quality. We condition the K1 model on Any4D tracks of increasing density (Tab. 12). Any4D reconstructs up to an unknown global scale, so we resolve one scale per video from ground-truth frame-0 depth; the target cameras are the benchmark trajectories. With estimated tracks, PSNR and SSIM decrease monotonically as density grows (19.34 dB at 1,000 tracks vs. 18.92 dB at 24,960), whereas with ground-truth tracks density has little effect (21.96–22.18 dB from 1,000 to 24,960 tracks, Tab. 8). In this setting, sparse conditioning is not a compromise: additional estimated tracks

bring more erroneous correspondences than useful signal. Together with Tab. 2, where reducing to 288 tracks hurts camera accuracy, this suggests that on the order of 10^3 tracks suffices: too few leave regions uncovered, while more estimated tracks mainly add errors.

Table 12: Generation with estimated (Any4D) tracks on Kubric-4D Moving-20 (81 frames, K1 model). Frame-0 and frame-80 queries are balanced.

Selection	Tracks	PSNR↑	SSIM↑	LPIPS↓
Uniform	1,000	**19.34**	**0.5256**	0.2854
Uniform	2,496	19.18	0.5206	**0.2829**
Uniform	6,240	19.00	0.5157	0.2846
Uniform	12,480	18.94	0.5149	0.2867
Uniform (full grid)	24,960	18.92	0.5141	0.2872

F Data Release

We will release the paired source/target track annotations used to train Track2View. Producing them with SpatialTrackerV2 on temporally concatenated view pairs (Sec. 3.3) took approximately 2,880 GPU-hours.