# MANIFOLD4D: DENOISING ON POINT CLOUD RENDERED MANIFOLDS FOR VIDEO RE-SHOOTING

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

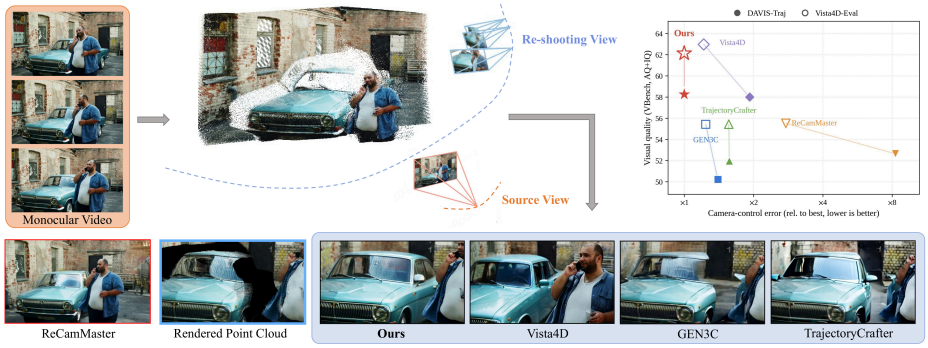


Figure 1: **Video re-shooting.** Given a monocular video and a target camera trajectory, MANIFOLD4D re-shoots the video along the new trajectory, achieving the best trajectory control among existing methods while preserving visual quality. The bottom panel shows a re-shooting example. GEN3C and TrajectoryCrafter inherit the artifacts of the imperfect render, while Vista4D leaves the subject misaligned with the render geometry. In the top-right plot, the vertical axis averages the VBench aesthetic and imaging quality scores; the horizontal axis denotes the camera-control error relative to the best method (see Appendix A for details).

## ABSTRACT

Video re-shooting is the task of re-rendering a monocular video of a dynamic scene along a user-specified camera trajectory. A dominant paradigm explicitly provides the target geometry by lifting the source video into a 4D point cloud using per-frame depth, and rasterizing this representation along the target trajectory to obtain point-cloud renders as geometric conditions for trajectory-controlled generation. The point cloud render and source video may provide conflicting visual cues throughout denoising, as both are directly used as conditions. We refer to this conflict as a trust dilemma, which can compromise trajectory control, 3D consistency of dynamic content, and visual quality, particularly under distribution shifts. We argue that because a point-cloud render is pixel-aligned with the target view, it does not need to be continuously provided as a conditioning signal throughout denoising. We propose MANIFOLD4D, which injects the render directly into the initial noise of flow matching, so that generation no longer departs from standard Gaussian noise but from a geometry-bearing noise distribution that we term the point cloud rendered manifold, leaving the source video as the only visual condition. The render is thus used exactly once, and the network is never asked to learn how to read it; in subsequent denoising steps the model can focus on the source video. In our experiments, MANIFOLD4D attains the best camera-control accuracy on every metric, lowering rotation error by 25–27% and translation error by up to 32% over the strongest baseline, while matching it in video fidelity and attaining the best overall photometric quality and 3D consistency of dynamic content.

## 1 INTRODUCTION

Camera motion is one of the oldest devices of visual storytelling: an orbiting shot or a well-timed dolly can turn an ordinary take into a cinematic one. *Video re-shooting*, camera-controlled video



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

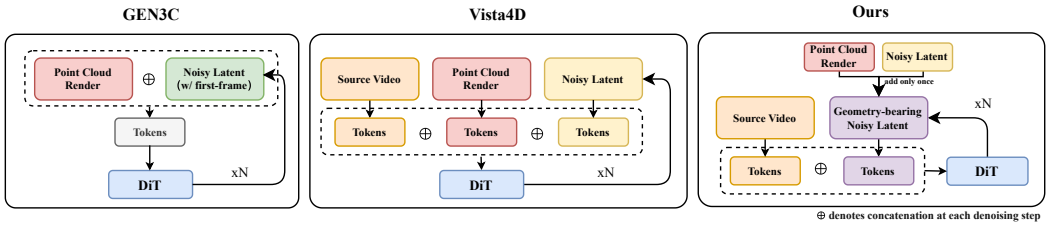


Figure 2: **How each method consumes the render.** Prior methods concatenate the render at every denoising step, so the network must learn to interpret and utilize it. Ours directly adds it into the starting noisy latent, requiring no learning. This explicit, one-time consumption is the key difference.

generation from a single video, brings this freedom to post-production. Given a monocular video of a dynamic scene and a user-specified camera trajectory, the goal is to render the same take from new viewpoints: reconstructing faithfully what the source video observes, synthesizing plausibly what it does not, and following the requested camera motion precisely. It also supplies a multi-view generative prior for monocular 4D reconstruction and gives world models an explicit handle on viewpoint motion.

To equip a pretrained video diffusion model with such camera-control, the camera trajectory has to be handed to the network in some form, and existing designs differ mainly in how explicit that form is. ReCamMaster (Bai et al., 2025a) keeps the trajectory implicit: the poses are compressed into an embedding, and the network is left to infer the implied geometry, which makes accurate control challenging. A more explicit line of work first lifts the source video into a point cloud using per-frame depth, then rasterizes it along the target trajectory to obtain rendered frames as geometric guidance, with the source video providing appearance and style guidance. GEN3C (Ren et al., 2025) and TrajectoryCrafter (Yu et al., 2025a) concatenate this render with the noisy latent at every denoising step, while the source video enters only weakly through a first-frame anchor or cross-attention. The render is often fragmented by sparse point clouds and distorted by depth errors, so these artifacts propagate into the output. Vista4D (Lin et al., 2026) instead balances the two as token streams of equal status, keeping visual quality and trajectory control intact, but adds no constraint enforcing the render geometry, so the subject can drift away from the target view. Fig. 1 illustrates both failure modes.

We refer to this tension as a trust dilemma between the source video and the point cloud render: which one to believe, and how much. Trusting the render too little weakens trajectory control and 3D consistency, while trusting it too much propagates its artifacts into the output. More fundamentally, this dilemma arises from asking the model to continuously reconcile two visual conditions, a balance that may not generalize beyond the motion magnitudes seen during training. A natural way to resolve this dilemma is to reduce their interaction during denoising. Independently and contemporaneously, MoCam (Liu et al., 2026) addresses this dilemma by assigning the render and source condition to different denoising stages. However, the render remains visible over multiple denoising steps, still requiring the network to learn how to interpret and utilize geometry that is already explicitly given. Despite depth errors and holes, the point cloud render is already in the target view and pixel-aligned with the video to be produced. We therefore argue that the network need not repeatedly learn how to interpret and utilize this explicit geometric information. Instead, we add the render to the initial noise, while the source video guides subsequent denoising. Fig. 2 contrasts the three designs in how the render enters denoising.

We propose MANIFOLD4D. The design philosophy is simple: remove the trust dilemma rather than balance the two signals. The two signals enter at different points and never compete during denoising. The render is added into the initial noise a single time, which displaces the starting distribution off the Gaussian prior and onto the point cloud rendered manifold. The source video enters at the token level and remains the only visual condition, so the network can spend its attention on appearance and detail rather than on weighing two competing signals, and retains ample room to correct an erroneous render. This design also reduces training cost, as the network no longer needs to learn how to interpret and utilize the render as a visual condition. It also infers about $1.5\times$ faster than Vista4D with the same base model and output format (Appendix E.1). We further show experimentally that such an injection does not destroy the diffusion prior: the model adapts to it after

fine-tuning. The simplest entry for the render turns out to be the strongest: trained with an identical budget, the natural alternatives that keep the render as a persistent condition fall behind in camera control (Sec. 4.4).

We conduct extensive experiments on multiple benchmarks. MANIFOLD4D ranks first on every camera-control metric of both benchmarks while matching the strongest baseline in visual quality, lowering rotation error by $25\%$ and $27\%$ over the best baseline on each and translation error by up to $32\%$, and it achieves clear advantages in trajectory following and 3D consistency of the dynamic subject in our user study. The advantage widens as the camera motion grows well beyond the training range, up to a $180^\circ$ change in viewing angle. We further report a robustness test in which the point cloud render contains clear errors, and show that our model is still able to correct them from the source video. Our contributions are summarized as follows:

- We analyze the trust dilemma faced by prior work and offer a new perspective: for pixel-aligned geometric priors, directly incorporating them into the starting state can be more effective than providing them as persistent conditions.
- We propose MANIFOLD4D, a video re-shooting model that injects the point cloud render into the starting point of flow matching so that generation departs from geometry-bearing noise and attains both visual quality and trajectory control.
- We conduct extensive experiments showing that MANIFOLD4D approaches or surpasses the pose-estimation accuracy of the point cloud render itself (treated as a geometric reference) while preserving visual quality and 3D consistency.

## 2 RELATED WORK

**Pose-conditioned video generation.** One line of work encodes the target trajectory into a numerical signal, such as extrinsic matrices or pixel-wise Plücker embeddings, and appends it to the conditioning stream, leaving the network to infer the implied geometry on its own. Where no source video is available, this yields camera-controlled text- and image-to-video models (Liu et al., 2023b; Wang et al., 2024; He et al., 2025; Bahmani et al., 2025). The same recipe carries over to video re-shooting, with the source video supplied as an additional reference (Van Hoorick et al., 2024; Bai et al., 2025a; Zhang et al., 2025; Luo et al., 2025; Jeong et al., 2025; Xie et al., 2024). In both settings a pose encoding states where the camera goes but not what it should see, so the mapping from camera parameters to pixels must be learned entirely from data, which yields weak trajectory control that cannot be previewed before generation.

**Render-conditioned video generation.** A second line of work makes the geometry explicit: per-frame depth lifts the source video into a point cloud, which is rasterized along the target trajectory into a point cloud render. This render is a far more explicit condition than a pose encoding, since it is pixel-aligned with the target view and states what should appear at every pixel. This pipeline was established for static scenes from single or sparse images (Chan et al., 2023; Wu et al., 2024; Yu et al., 2025b), then carried to monocular video by TrajectoryCrafter (Yu et al., 2025a) and GEN3C (Ren et al., 2025). Both concatenate the render channel-wise with the noisy latent, and differ only in the appearance reference: the whole source video in the former, its first frame alone in the latter. EX-4D (Hu et al., 2025) instead lifts the source scene into a watertight mesh for extreme viewpoint 4D synthesis, while Vista4D (Lin et al., 2026) builds a 4D point cloud and concatenates render, source video and the noisy latent at the token level. Since the render and the source video are both explicit visual conditions, they readily compete. Independently and contemporaneously with our work, MoCam (Liu et al., 2026) identifies this conflict and separates the two signals in time, handing over from render to source video partway through denoising. Scheduled or static, all methods above ask the network to learn to read a signal that is already given. We instead inject the render into the initial noise, so that geometry is carried by the generative state rather than repeatedly supplied as an explicit conditioning stream, with fine-tuning that matches the resulting distribution.

**Signal injection as a starting point.** SDEdit (Meng et al., 2022) introduces the paradigm of adding noise to an existing signal and using the result as the starting point for diffusion, rather than supplying the signal as a condition; the principle generalizes beyond Gaussian noise (Bansal et al., 2023) and applies to inpainting (Lugmayr et al., 2022). Image restoration exploits the same

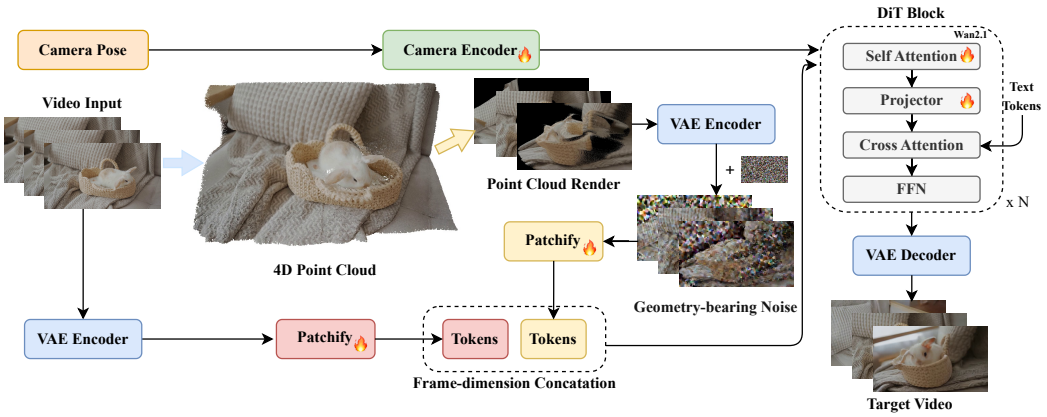


Figure 3: **Pipeline overview.** Given a monocular video and a target camera trajectory, MANIFOLD4D takes a 4D point cloud obtained from an existing reconstruction and motion decomposition pipeline and renders target-trajectory views from it. The rendered views are encoded into VAE latents, which are injected into the noise and patchified into tokens. The source video is similarly encoded into tokens, concatenated with the noise tokens, and fed to the DiT for denoising, together with the encoded camera trajectory.

idea: initializing the trajectory from the degraded observation shortens the generative path and improves fidelity (Chung et al., 2022; Delbracio & Milanfar, 2023; Liu et al., 2023a). Across different models and tasks, these works establish signal injection as an effective starting point; we carry this idea into video generation and propose MANIFOLD4D, which injects the point cloud render into the starting point of flow matching, realizing a new approach to video re-shooting.

# 3 METHOD

## 3.1 FLOW MATCHING FOR VIDEO DIFFUSION

We build on Wan2.1-T2V (Wan et al., 2025), a latent video diffusion transformer trained with flow matching (Lipman et al., 2023). A VAE first compresses the video spatially and temporally, and the velocity field is learned in this latent space.

Flow matching learns a velocity field that transports a prior distribution to the data distribution along a straight path. We follow the convention that $t=0$ is clean data and $t=1$ is the prior, so for a target latent $x_{\text{tgt}}$ and Gaussian noise $\varepsilon \sim \mathcal{N}(0, I)$ the training path and its velocity are

$$x_t = (1-t)\, x_{\text{tgt}} + t\, \varepsilon, \qquad v^\star = \varepsilon - x_{\text{tgt}}, \tag{1}$$

and the network $v_\theta$ is trained to regress $v^\star$ under $\mathbb{E}_{t, x_{\text{tgt}}, \varepsilon} \| v_\theta(x_t, t, c) - v^\star \|^2$ where $c$ denotes the conditioning input. For a standard text-to-video model, $c$ is solely the text prompt, while trajectory-controlled variants place both the source video and the render in $c$: the trust dilemma analyzed in Sec. 1. We fine-tune Wan2.1-T2V differently: the render is injected into the starting point of the path at $t{=}1$, while the source video remains the only visual condition.

## 3.2 EXPLICIT 4D POINT CLOUD

Following the explicit-control paradigm, we first reconstruct a 4D point cloud and project it along the target trajectory to obtain a rendered video $x_{\text{render}}$. Specifically, we estimate the per-frame depth $D_i$, camera intrinsics $K_i$, and extrinsics $T_i = [R_i \mid \mathbf{t}_i]$ using VGGT-Omega (Wang et al., 2026a), an extension of VGGT (Wang et al., 2025), and back-project every pixel of every source frame into a globally aligned point cloud.

To enable 4D reconstruction, we further segment the dynamic subject using Qwen2.5-VL (Bai et al., 2025b) and SAM3 (Carion et al., 2026) to obtain a motion mask that separates dynamic points from the static background. Static points are accumulated over all source frames into one global map $\mathcal{P}_{\text{static}}$, whereas dynamic points are kept per timestamp, $\mathcal{P}^{(i)}_{\text{dyn}}$, because their world position changes

over time. Rendering target frame $j$ therefore rasterizes $\mathcal{P}_{\text{static}} \cup \mathcal{P}_{\text{dyn}}^{(j)}$ through the target camera $(K_j, T_j)$, yielding the rendered video $x_{\text{render}}$. Rasterization also produces a binary mask indicating whether each pixel receives at least one point. We average-pool this mask to the token resolution to obtain the coverage $\alpha \in [0, 1]$ for each token, which serves as an indicator for distinguishing regions that require generation from those that only need refinement.

### 3.3 Geometry-bearing starting point

The 4D point cloud provides the geometric structure, while the source video serves as the reference for appearance, style, and fine-grained details. The two therefore enter the model in different ways: geometry initializes generation, and appearance guides denoising.

Unlike conventional approaches that concatenate the source video, noise, and point-cloud rendering as three token streams, we inject the render into the noise to obtain a geometry-bearing noise representation:

$$\tilde{x}_1 = x_{\text{render}} + \sigma\,\varepsilon, \qquad x_1 = \alpha\,\tilde{x}_1 + (1-\alpha)\,\varepsilon, \tag{2}$$

where $\varepsilon \sim \mathcal{N}(0, I)$ is standard Gaussian noise, $\tilde{x}_1$ is the starting state of covered tokens, i.e., the render plus a residual noise of strength $\sigma$, and $\alpha$ the per-token coverage defined above. Hole tokens therefore start from pure Gaussian noise, fully covered tokens from $x_{\text{render}} + \sigma\,\varepsilon$, and partial coverage interpolates the two. Training follows Eq. equation 1 with the Gaussian endpoint replaced by this geometry-bearing noise, so the regressed velocity becomes $v^\star = x_1 - x_{\text{tgt}}$. The objective is otherwise unchanged: $v_\theta$ regresses $v^\star$ on all tokens, transporting the point cloud rendered manifold, rather than the Gaussian prior, to the data distribution. Because diffusion models lay down geometric structure in their early steps, the explicit injection lets generation start with this skeleton already in place, freeing the network's capacity for appearance and detail.

We then concatenate the geometry-bearing noise with the source video at the token level, $\mathbf{z} = [\,\mathbf{x}_1\,;\,\text{Patchify}(\text{Enc}(S))\,]$, where $\text{Enc}(S)$ is the VAE-encoded source-video latent.

Because the render is dropped for part of training, we additionally supply the target trajectory as per-pixel Plücker embeddings through zero-initialized linear projections (Bai et al., 2025a). Removing them at inference leaves accuracy and quality essentially unchanged (Table 5); we retain them as a training-time safeguard.

Our key insight is that explicit geometry is most effective at the starting point of the generative path rather than as a persistent condition: since the render is pixel-aligned with the target view, consuming it there exploits this alignment directly, and a persistent render condition would also compete with the source video condition. One may worry that such an injection breaks the diffusion prior; our experiments show that the model adapts to this shift through fine-tuning.

### 3.4 Training detail

We build on Wan2.1-T2V-14B and fine-tune patchify layers, self-attention layers, camera encoders, and projectors, while freezing all other parameters. We use AdamW with a constant learning rate of $10^{-5}$ and a global batch size of 8 for 30K steps, at 49 frames and $384 \times 672$.

**Dataset.** Training mixes five sources (DL3DV (Ling et al., 2024), DynPose (Rockwell et al., 2025), OpenVid-HD (Nan et al., 2025), MultiCamVideo (Bai et al., 2025a), and HuMMan (Cai et al., 2022); about 36K clips), with depth and camera poses from VGGT-Omega for every source except HuMMan, which ships with calibrated captures; further details in Appendix B.

## 4 Experiments

### 4.1 Experimental setup

**Benchmarks.** We select 24 dynamic scenes from DAVIS (Pont-Tuset et al., 2017) to form our benchmark, DAVIS-Traj. Each scene is re-shot under three trajectory patterns that orbit the dynamic subject while keeping it on the optical axis, sweeping up to $\pm 60^\circ$ of yaw with additional pitch and distance variations, giving 72 evaluation clips; each clip is 49 frames long, and Fig. 6 in Appendix C

Table 1: **Camera control accuracy.** MANIFOLD4D attains the highest accuracy on every metric, approaching or surpassing the point cloud render itself on rotation error. **Bold** denotes the best and underline the second-best result.

| Method | DAVIS-TRAJ (72 clips) | | | Vista4D-Eval (103 clips) | | | Vista4D-Eval raw (110 clips) | | |
|---|---|---|---|---|---|---|---|---|---|
| | RotErr ↓ | TransErr ↓ | IntrErr ↓ | RotErr ↓ | TransErr ↓ | IntrErr ↓ | RotErr ↓ | TransErr ↓ | IntrErr ↓ |
| ReCamMaster | 14.04 | 0.581 | 2.77 | 11.81 | 0.495 | 10.63 | 13.73 | 0.549 | 11.23 |
| TrajectoryCrafter | 1.56 | 0.274 | 1.14 | 5.40 | 0.462 | 8.40 | 8.24 | 0.518 | 9.14 |
| GEN3C | 1.45 | 0.236 | 1.13 | 3.77 | 0.441 | 8.74 | 4.31 | 0.460 | 9.33 |
| Vista4D | 2.56 | 0.240 | 1.47 | 3.89 | 0.403 | 7.66 | 4.84 | 0.432 | 8.10 |
| OURS | **1.09** | **0.161** | **1.04** | **2.76** | **0.394** | **7.65** | **4.26** | **0.424** | **8.09** |
| Point cloud render | *1.09* | *0.094* | *1.02* | *3.59* | *0.328* | *6.91* | *4.18* | *0.390* | *7.20* |

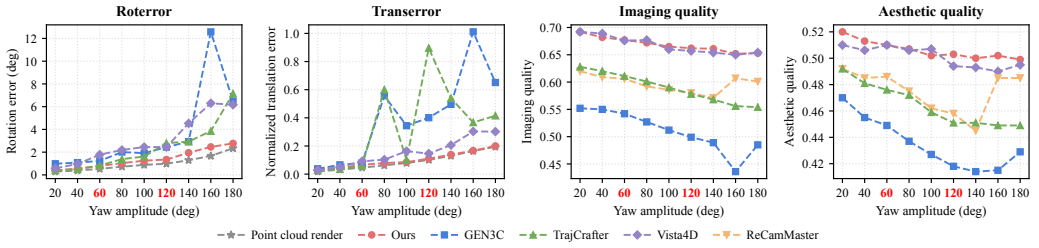


Figure 4: **Control error and visual quality under growing motion amplitude.** MANIFOLD4D maintains the lowest pose error at almost every amplitude, staying close to the point cloud render, while other methods degrade markedly. Visual quality (Aesthetic and Imaging) likewise remains the highest, especially at large amplitudes. Red tick labels mark the amplitudes of DAVIS-TRAJ.

visualizes the three patterns and their exact parameterization. We also select ten of these scenes to construct a dataset of growing motion amplitudes, probing the limits of model capability. We additionally report on the 110 evaluation clips released with Vista4D (Lin et al., 2026), and on the real iPhone multi-view dataset (Gao et al., 2022).

**Baselines.** We compare against four published methods spanning both ways of supplying camera control: ReCamMaster (Bai et al., 2025a), which conditions on an implicit camera embedding without explicit geometry; TrajectoryCrafter (Yu et al., 2025a) and GEN3C (Ren et al., 2025), which concatenate a point-cloud render along the channel axis; and Vista4D (Lin et al., 2026), which concatenates it at the token level. For fairness, every method with explicit geometric control receives the same 4D point-cloud projection at inference time. Each method otherwise runs under its own best default settings, and all outputs are resampled to $384 \times 672$ and 49 frames for evaluation. ReCamMaster and GEN3C output more than 49 frames and require coincident first-frame poses of the source and target trajectories, so we prepend a transition segment from the source start to the target start and evaluate the subsequent 49 frames, which follow the target trajectory. The base model of every method and its per-clip inference time are reported in Appendix E.1.

**Point cloud render.** The point cloud render rasterizes the source point cloud along the target trajectory, with no generation involved. We use VGGT-Omega both to reconstruct the point cloud from the source video and, for evaluation, to estimate poses on the source and each generated video jointly. We also evaluate the trajectory of the point cloud render itself to quantify the systematic error of the estimator and our alignment protocol; the render is excluded from ranking and, since its holes introduce some error, treated as a reference rather than strict ground truth.

### 4.2 QUANTITATIVE COMPARISONS

**Camera control accuracy.** We measure rotation, translation, and intrinsic errors on both benchmarks, DAVIS-Traj and Vista4D-Eval; the alignment protocol and metric definitions are given in Appendix D. Because some Vista4D-Eval clips have only a small overlap between the source and target trajectories, coordinate alignment goes conspicuously wrong on them, so for fairness we additionally report results after removing clips with a rotation error above $60^\circ$ in any method or seed. In

Table 2: **Novel view photometric quality on iPhone Dataset.** MANIFOLD4D achieves state-of-the-art performance across key photometric and optical-flow metrics.

| Method | PSNR ↑ | SSIM ↑ | LPIPS ↓ | mPSNR ↑ | mSSIM ↑ | mLPIPS ↓ | EPE ↓ |
|---|---|---|---|---|---|---|---|
| ReCamMaster | 10.89 | 0.3363 | 0.7161 | 11.00 | 0.3326 | 0.5376 | 6.66 |
| TrajectoryCrafter | 15.52 | **0.4338** | 0.5162 | 16.41 | **0.4476** | 0.3494 | 1.58 |
| GEN3C | 14.75 | 0.3937 | 0.4609 | 15.87 | 0.4119 | 0.3103 | 1.47 |
| Vista4D | 15.78 | 0.3913 | 0.3884 | 16.30 | 0.3997 | 0.2600 | 1.28 |
| **OURS** | **16.21** | 0.3847 | **0.3582** | **16.65** | 0.3887 | **0.2393** | **1.25** |

Table 3: **Visual quality.** FID/FVD are computed against the corresponding source clips; Aesthetic and Imaging are VBench scores ($\times 100$).

| Method | DAVIS-TRAJ | | | | Vista4D-Eval | | | |
|---|---|---|---|---|---|---|---|---|
| | FID ↓ | FVD ↓ | Aesthetic ↑ | Imaging ↑ | FID ↓ | FVD ↓ | Aesthetic ↑ | Imaging ↑ |
| ReCamMaster | 65.55 | **1122.9** | 48.099 | 57.204 | **92.79** | **1181.1** | 52.338 | 58.694 |
| TrajectoryCrafter | 85.27 | 1571.6 | 46.899 | 57.038 | 116.31 | 1581.4 | 47.579 | 63.300 |
| GEN3C | 89.76 | 1463.9 | 47.245 | 53.218 | 113.43 | 1534.3 | 48.852 | 62.001 |
| Vista4D | **55.64** | 1208.2 | 51.096 | 64.914 | 98.12 | 1404.6 | **55.012** | **70.933** |
| OURS | 57.55 | 1222.7 | **51.138** | **65.384** | 97.37 | 1360.2 | 53.757 | 70.509 |

Table 1, MANIFOLD4D keeps the lowest rotation and translation errors in both the filtered and unfiltered settings, and on rotation even approaches or surpasses the render itself. The render is weaker on Vista4D-Eval because some scenes contain large empty regions that degrade the pose estimator. To check that the conclusions do not depend on the pose estimator, we repeat the pose estimation and alignment with Pi3 (Wang et al., 2026b), and MANIFOLD4D again performs best overall; see Appendix E.2. All metrics are averaged over three seeds to reduce random estimation error.

**Generalization to larger camera motions.** We select ten of the benchmark scenes and, for each, construct a family of trajectories orbiting the dynamic subject, with a per-side motion amplitude of $10^\circ$ to $90^\circ$ in yaw (total sweep $20^\circ$ to $180^\circ$). We evaluate all methods on this family, together with the point cloud render as a reference. As Fig. 4 shows, MANIFOLD4D keeps the lowest rotation error at almost every amplitude, and the lowest translation error from moderate amplitudes onward, whereas the error of every other method rises markedly. MANIFOLD4D also maintains visual quality at large amplitudes, showing that our explicit design stays stable at amplitudes unseen during training, while other methods begin to degrade. ReCamMaster's pose error is too large to plot alongside the other methods. The complete numerical results are reported in Tables 8 and 9 of Appendix E.3. We recommend watching the supplementary demo video for visual results at the largest amplitudes.

**Novel-view photometric quality.** We evaluate photometric quality and 3D consistency on the real-world time-synchronized multiview dataset, iPhone (Gao et al., 2022). Table 2 reports PSNR, SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018) and their masked variants (mPSNR / mSSIM / mLPIPS) over the coverage mask of the point cloud render (Sec. 4.1), together with end-point error (EPE) from RAFT (Teed & Deng, 2020) optical flow, which measures how closely the generated video follows the ground-truth motion. MANIFOLD4D performs best on all metrics except SSIM and mSSIM, where TrajectoryCrafter leads. However, viewing the synthesized videos quickly reveals significant artifacts in TrajectoryCrafter's outputs that SSIM fails to capture. We show examples in Fig. 7 of Appendix F.

**Video fidelity.** We evaluate video fidelity and quality on both benchmarks, DAVIS-Traj and Vista4D-Eval, reporting FID (Heusel et al., 2017), FVD (Unterthiner et al., 2018) and VBench (Huang et al., 2024) scores (aesthetic quality and imaging quality) in Table 3; the remaining metrics are reported in Appendix E.4. MANIFOLD4D is on par with Vista4D across the board, which means the tight trajectory control does not come at the cost of visual quality through copying the render. ReCamMaster scores best on FVD on both benchmarks (and on FID on Vista4D-Eval), which we attribute to its tendency to ignore the target trajectory and stay close to the source content. The low quality of GEN3C and TrajectoryCrafter stems from over-reliance on the render, which fragments and blurs the dynamic content. We show examples in Fig. 5.




Figure 5: **Qualitative results on monocular input.** Dashed guides are placed on the point cloud render and copied to every row: green marks the expected position of the dynamic subject, red a static background structure. Our method shows stable control and pleasing visual quality while Vista4D's dynamic subject drifts away from the green guide. GEN3C and TrajectoryCrafter inherit the sparsity of the render, leaving subjects fragmented; ReCamMaster stays clean but ignores the requested view.

Table 4: **User study.** Selection rate in %. MANIFOLD4D is preferred by large margins on trajectory following and 3D consistency of the dynamic subject, and remains competitive on overall quality.

| Criterion | ReCamMaster | TrajectoryCrafter | GEN3C | Vista4D | OURS |
|---|---|---|---|---|---|
| Trajectory following | 1.4% | 17.0% | 20.5% | <u>28.3%</u> | **63.6%** |
| 3D consistency | 1.8% | <u>15.6%</u> | 9.5% | 8.7% | **62.4%** |
| Overall quality | 12.9% | 3.2% | 5.8% | **54.7%** | <u>49.6%</u> |

### 4.3 QUALITATIVE COMPARISONS

**Geometric alignment of dynamic content.** In Fig. 5, the red and green dashed guides track the static background and the expected position of the dynamic subject, respectively, probing 3D consistency on both. Vista4D keeps the static background aligned with the render, but its dynamic subject drifts away from the green guide. This behavior also explains its low trajectory error: for pose estimation, only the static background provides reliable geometric cues. GEN3C and TrajectoryCrafter are similarly consistent, but when sparsity lets the background leak through the point cloud, they fail to complete the missing content and leave the subject fragmented. ReCamMaster, in contrast, loses geometric control altogether. Taken together, MANIFOLD4D simultaneously delivers precise camera control, dynamic-object consistency, and visual quality, and exhibits the strongest stability at the boundary of its capability. We also provide a supplementary demo video with additional examples.

**User study.** We present the results of our user study in Table 4, where we ask users to evaluate all methods in three aspects: trajectory following, 3D consistency of the dynamic subject, and visual quality (multiple selections are permitted). Twenty clips, ten sampled from each of the two benchmarks, are rated by over 30 participants, who do not know the generating model. The results show MANIFOLD4D achieves a clear advantage in trajectory following and 3D consistency of the dynamic subject, while ranking second on visual quality with a narrow margin to Vista4D. To help participants judge objectively, we provide an optional overlay of the point cloud render, making geometric misalignment directly visible; the evaluation interface is shown in Appendix G.

**Robustness to imperfect geometry.** To test how MANIFOLD4D copes with a degraded point cloud, we remove the motion mask at render time so that all dynamic points are stacked together

Table 5: **Inference ablation** on Vista4D-Eval. Pose errors on the 103-scene filtered subset; quality averaged over all 110 scenes. Left: conditioning ablation. Middle: residual noise strength $\sigma$ sweep. Right: architecture ablation; all variants share the same base model, seed, and early-stage training budget.

**(a) Conditioning ablation**

| Setting | Camera | | Quality | |
|---|---|---|---|---|
| | Rot↓ | Trans↓ | AQ↑ | IQ↑ |
| OURS (full) | 2.79 | 0.374 | 54.0 | 70.5 |
| w/o camera emb. | **2.62** | **0.370** | 54.1 | 70.5 |
| w/o text prompt | 3.86 | 0.458 | 50.0 | 70.0 |
| w/o source video | 13.07 | 0.636 | 50.4 | 67.3 |
| w/o injected render | 34.11 | 0.618 | **56.5** | **71.9** |

**(b) Residual noise strength $\sigma$**

| $\sigma$ | Camera | | Quality | |
|---|---|---|---|---|
| | Rot↓ | Trans↓ | AQ↑ | IQ↑ |
| 0.0 | **2.71** | 0.397 | 53.8 | 69.5 |
| 0.3 | 2.79 | **0.374** | **54.0** | 70.5 |
| 0.6 | 2.90 | 0.407 | 53.5 | 71.0 |
| 0.9 | 3.89 | 0.461 | 52.3 | **71.4** |
| 1.2 | 5.73 | 0.545 | 47.3 | 69.4 |
| 1.5 | 8.05 | 0.645 | 40.0 | 64.9 |

**(c) Architecture ablation**

| Architecture | Camera | | Quality | |
|---|---|---|---|---|
| | Rot↓ | Trans↓ | AQ↑ | IQ↑ |
| OURS | **3.10** | **0.391** | 53.7 | 69.6 |
| Vista4D arch | 3.33 | 0.438 | 54.2 | **70.8** |
| GEN3C arch | 4.96 | 0.448 | **54.7** | 69.0 |

across frames, producing a render with incorrect dynamic geometry. Despite the corrupted render, the model still recovers the correct motion from the source video. This highlights the benefit of injecting the render only once: errors in the render can be corrected from the source video during denoising. Visualizations are shown in Fig. 8, Appendix F.2.

### 4.4 ABLATION STUDY

**Inference-time ablation.** Table 5(a) removes each conditioning stream at inference time. Removing the camera embedding leaves trajectory control essentially unchanged. The slight pose gain is within the noise of single-seed evaluation rather than a genuine improvement. Removing the text prompt causes a mild degradation. The source video and the point cloud render are both critical: suppressing the source video degrades rotation by $4.7\times$, and suppressing the render collapses it by $12\times$. Note that *w/o injected render* achieves the highest visual quality, as it degenerates into replicating the source video. Table 5(b) sweeps the residual noise strength $\sigma$ in Eq. 2, the extra noise overlaid on the render at the starting point, from the noise-free render up to $5\times$ the default. Moderate residual noise improves imaging quality, while excessive noise degrades both control and quality. The noise-free variant ($\sigma$=0) retains the strongest camera control but lower visual quality. Starting directly from the raw render leaves the model no room to correct it, while too much noise drowns out the geometric prior. Moderate noise instead provides this freedom, so the model can refine the render and recover the details it misses. We adopt $\sigma$=0.3 as the default, which attains the best translation accuracy and aesthetic quality.

**Architecture variants.** To separate the contributions of our architecture and our training recipe, we train the conditioning layouts of the two strongest baselines with the same base model, seed and early-stage training budget. In Table 5(c), both variants fall behind MANIFOLD4D on camera control, with visual quality comparable across the three. The Vista4D-style variant trades pose accuracy for a small imaging-quality gain, and also shows lingering geometric inconsistency on dynamic objects (milder than the original Vista4D) and about $1.5\times$ the inference time. The GEN3C-style variant shows visible quality degradation in some scenes despite a slightly higher aesthetic score, and its rotation error is $1.6\times$ ours. The advantage of MANIFOLD4D therefore comes from architecture and training recipe together. In this controlled comparison, the simplest layout wins. Appendix F.3 visualizes the three layouts side by side, making the artifact patterns directly visible.

## 5 CONCLUSION

We have presented MANIFOLD4D, which re-shoots a monocular video along a user-specified camera trajectory. By injecting explicit geometry into the noise once and keeping the source video as the only visual condition, MANIFOLD4D resolves the competition between conditions and uses the geometric prior directly. Extensive experiments show that this simple design suffices: MANIFOLD4D attains the best trajectory control on every benchmark with competitive visual quality, stays geometrically consistent on dynamic objects and reliable under large camera motions.

**Limitations and future work.** We believe the generative prior of the pretrained diffusion model still holds potential that more sophisticated noise–geometry interactions could unlock. Designing such interactions is a promising direction for future work.

## REPRODUCIBILITY STATEMENT

We build MANIFOLD4D entirely on publicly available models and datasets. Sec. 3 describes the architecture and the training objective. Appendix B lists the base model, the trainable subset, the five training data sources with their sampling weights, all optimization hyperparameters, and all training techniques, so the full training procedure can be reproduced. Appendix C documents the construction of DAVIS-TRAJ, including the scene selection and the parameterization of the three trajectory patterns. We will release the benchmark to support reproduction of our results. Appendix D provides the closed-form alignment procedure, the filtering rule, and the exact definitions of all pose metrics. We will release the inference and evaluation code and the trained weights publicly. Note that the inference code is already included in the supplementary material.

## ETHICS STATEMENT

MANIFOLD4D re-shoots a monocular video along a new camera trajectory. Like other camera-controllable video generation methods, it could in principle be misused to re-frame real recordings in ways that mislead viewers about how the footage was captured. Two properties reduce this risk. First, MANIFOLD4D creates no new subjects: the subject, its appearance, and the depicted events come from the source video, so the method changes the viewpoint rather than the depicted content. Second, all training and evaluation data come from datasets that are publicly released for research purposes, and we collected no data ourselves. We will release the code and weights for research use upon publication, and we encourage users to respect the terms of their source material and the content policies of their platforms.

## AI USE STATEMENT

In this work, we used generative AI tools to assist with code development (including data cleaning and visualization scripts) and with translating and polishing the language of the manuscript. We have not used generative AI tools to develop the research methodology, formulate hypotheses, design the model, or interpret the experimental results. We have reviewed and verified all AI-assisted work. We take responsibility for the final content of this work, including text, claims, code, data processing, and visualizations produced with the aid of generative AI.

## REFERENCES

Sherwin Bahmani, Ivan Skorokhodov, Aliaksandr Siarohin, Willi Menapace, Guocheng Qian, Michael Vasilkovsky, Hsin-Ying Lee, Chaoyang Wang, Jiaxu Zou, Andrea Tagliasacchi, et al. VD3D: Taming large video diffusion transformers for 3D camera control. In *International Conference on Learning Representations*, 2025.

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie Li, Hamed Kazemi, Furong Huang, Micah Goldblum, and Jonas Geiping. Cold diffusion: Inverting arbitrary image transforms without noise. In *International Conference on Learning Representations*, 2023.

Zhongang Cai, Daxuan Ren, Ailing Zeng, Zhengyu Lin, Tao Yan, Wenjia Wang, Xiangyu Zhang, Yifan Wei, Xiaoyi Zhu, Zhengzhe Zhang, Mingyuan Zhang, Lei Yang, Ziwei Liu, and Dahua Lin. HuMMan: Multi-modal 4D human dataset for versatile sensing and modeling. In *European Conference on Computer Vision*, 2022.

Nicolas Carion, Laura Gustafson, Yuan-Ting Hu, Shoubhik Debnath, Ronghang Hu, Didac Suris, Chaitanya Ryali, Kalyan Vasudev Alwala, Haitham Khedr, Andrew Huang, Jie Lei, Tengyu Ma, Baishan Guo, Arpit Kalla, Markus Marks, Joseph Greer, Meng Wang, Peize Sun, Roman Rädle, Triantafyllos Afouras, Effrosyni Mavroudi, Katherine Xu, Tsung-Han Wu, Yu Zhou, Liliane Momeni, Rishi Hazra, Shuangrui Ding, Sagar Vaze, Francois Porcher, Feng Li, Siyuan Li, Aishwarya Kamath, Ho Kei Cheng, Piotr Dollár, Nikhila Ravi, Kate Saenko, Pengchuan Zhang, and Christoph Feichtenhofer. SAM 3: Segment anything with concepts. In *International Conference on Learning Representations*, 2026.

Eric R Chan, Koki Nagano, Matthew A Chan, Alexander W Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3D-aware diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4194–4206. IEEE, 2023.

Hyungjin Chung, Byeongsu Sim, and Jong Chul Ye. Come-closer-diffuse-faster: Accelerating conditional diffusion models for inverse problems through stochastic contraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Mauricio Delbracio and Peyman Milanfar. Inversion by direct iteration: An alternative to denoising diffusion for image restoration. *Transactions on Machine Learning Research*, 2023.

Hang Gao, Ruilong Li, Shubham Tulsiani, Bryan Russell, and Angjoo Kanazawa. Monocular dynamic view synthesis: A reality check. In *Advances in Neural Information Processing Systems*, 2022.

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

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017.

Tao Hu, Haoyang Peng, Xiao Liu, and Yuewen Ma. EX-4D: EXtreme viewpoint 4D video synthesis via depth watertight mesh. *arXiv preprint arXiv:2506.05554*, 2025.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 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, October 2025.

Kuan Heng Lin, Zhizheng Liu, Pablo Salamanca, Yash Kant, Ryan Burgert, Yuancheng Xu, Koichi Namekata, Yiwei Zhao, Bolei Zhou, Micah Goldblum, Paul Debevec, and Ning Yu. Vista4D: Video reshooting with 4D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32671–32682, June 2026.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, Xuanmao Li, Xingpeng Sun, Rohan Ashok, Aniruddha Mukherjee, Hao Kang, Xiaohui Kordjamshidi, Yu-Xiong Wang, Nathan Jacobs, Kyle Kim, Bedrich Zhou, and Bedrich Benes. DL3DV-10K: A large-scale scene dataset for deep learning-based 3D vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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*, 2023.

Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. $I^2$SB: Image-to-image Schrödinger bridge. In *International Conference on Machine Learning*, 2023a.

Haofeng Liu, Yang Zhou, Ziheng Wang, Zhengbo Xu, Zhan Peng, Jie Ma, Jun Liang, Shengfeng He, and Jing Li. MoCam: Unified novel view synthesis via structured denoising dynamics. In *European Conference on Computer Vision*, 2026.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3D object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9298–9309. IEEE, 2023b.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

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.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. SDEdit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022.

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*, 2025.

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. The 2017 DAVIS challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, volume 139, pp. 8748–8763, 2021.

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*, 2025.

Chris Rockwell, Joseph Tung, Tsung-Yi Lin, Ming-Yu Liu, David F. Fouhey, and Chen-Hsuan Lin. Dynamic camera poses and where to find them. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Zachary Teed and Jia Deng. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision*, 2020.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

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*, 2024.

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

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. *arXiv preprint arXiv:2605.15195*, 2026a.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. $\pi^3$: Permutation-equivariant visual geometry learning. In *International Conference on Learning Representations*, 2026b.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004.

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, 2024.

Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. ReconFusion: 3D reconstruction with diffusion priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21551–21561, 2024.

Yiming Xie, Chun-Han Yao, Vikram Voleti, Huaizu Jiang, and Varun Jampani. SV4D: Dynamic 3D content generation with multi-frame and multi-view consistency. *arXiv preprint arXiv:2407.17470*, 2024.

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, October 2025a.

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. ViewCrafter: Taming video diffusion models for high-fidelity novel view synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025b. doi: 10.1109/TPAMI.2025.3613256.

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, June 2025.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586–595, June 2018. doi: 10.1109/CVPR.2018.00068.

## A TEASER FIGURE DETAILS

In the teaser (Fig. 1), each method is plotted on both benchmarks from its metrics averaged over three seeds. The horizontal axis gives the camera-control error relative to the best method: RotErr and TransErr, computed on the common scene set, are each divided by the lowest value among the compared methods on that benchmark, and the two ratios are averaged, so the best method sits at 1; the axis uses a base-2 logarithmic scale. The vertical axis averages the two VBench scores, aesthetic and imaging quality, from Table 3 ($\times 100$).

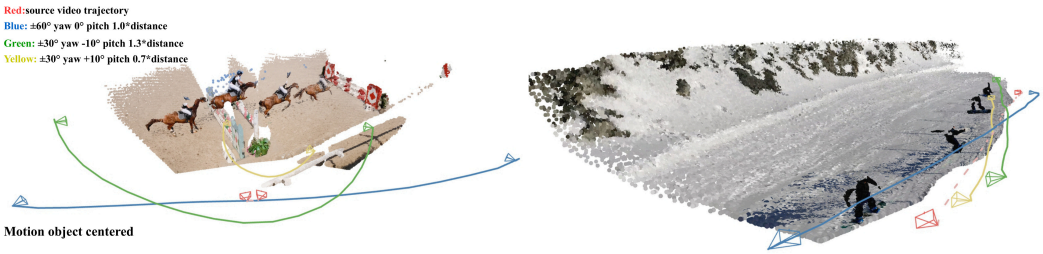


Figure 6: **Trajectory patterns of DAVIS-TRAJ.** Each pattern orbits the dynamic subject while keeping it on the optical axis. f1 (blue) sweeps $\pm 60^\circ$ of yaw; f2 (green) sweeps $\pm 30^\circ$ of yaw with $-10^\circ$ of pitch at $1.3\times$ distance; f3 (yellow) sweeps $\pm 30^\circ$ of yaw with $+10^\circ$ of pitch while dollying in from $1.0$ to $0.7$. The red curve shows the source camera trajectory.

## B TRAINING DETAILS

**Base model and trainable subset.** We build on Wan2.1-T2V-14B together with its native 16-channel VAE, which downsamples by $8\times$ spatially and $4\times$ temporally. Clips are 49 frames at $384 \times 672$, mapped by the VAE to 13 latent frames. Only the self-attention projections (Q/K/V/O with their RMSNorm layers), the four patchify embeddings, the per-block projectors applied to the self-attention output, and the per-block Plücker camera encoder are trainable; feed-forward blocks, cross-attention, normalization layers, time and text embeddings, the output head and all modulation parameters remain frozen. The camera encoder is zero-initialized, so at step $0$ the network is bit-identical to the pretrained backbone and the geometric pathway grows in from zero.

**Training data.** We train in a single mixed stage over five sources: DL3DV (Ling et al., 2024) (8.5k static scenes), DynPose (Rockwell et al., 2025) (8.0k) and OpenVid-HD (Nan et al., 2025) (14.9k) monocular dynamic clips, MultiCamVideo (Bai et al., 2025a) (3.2k synthetic scenes with ten cameras each, captured at three focal lengths), and HuMMan (Cai et al., 2022) (0.9k real ten-camera human captures). In our experience DL3DV and HuMMan matter comparatively little for final quality. Sampling weights give MultiCamVideo $56\%$, the monocular dynamic clips $22\%$, DL3DV $11\%$ and HuMMan $11\%$ of each epoch. Metric depth and camera poses come from VGGT-Omega (Wang et al., 2026a) for every source except HuMMan, which ships with calibrated multi-camera captures. The monocular clips carry no second view, so they are used through a double-reprojection cycle (Yu et al., 2025a): a synthetic target camera is sampled (up to $15^\circ$ yaw and $0.30$ of the scene extent in translation, rejected unless the warp retains at least $45\%$ coverage), the source is reprojected into it and back, and the round trip supplies a supervised pair.

**Optimization.** We train for 30k steps with AdamW at a constant learning rate of $10^{-5}$, batch size $1$ per GPU on eight A100s. Each conditioning stream (source, point cloud, camera, prompt) is dropped independently with probability $10\%$; dropping the point cloud sets the coverage $\alpha$ to zero, so the starting state falls back to pure Gaussian noise and the sample reverts to standard flow matching, and a further $5\%$ of samples are trained fully unconditionally to preserve the text-to-video prior. To prevent the model from copying the warp where it is unreliable, we augment the point cloud at render time: a slab of the dynamic subject is carved out with probability $20\%$, and static points are removed in spatial blocks or along temporal spans, each with probability $20\%$. Inspired by Vista4D (Lin et al., 2026), MultiCamVideo and HuMMan additionally use $50\%$ temporal reversal.

## C BENCHMARK CONSTRUCTION DETAILS

**Scene selection and trajectory patterns.** We select 24 dynamic scenes from DAVIS (Pont-Tuset et al., 2017) and re-shoot each of them under three trajectory patterns (f1–f3), giving 72 evaluation clips; a representative sample is visualized in Fig. 6. We additionally report on the 110 evaluation clips released with Vista4D (Lin et al., 2026), so that every method is scored on both a purpose-built and a third-party benchmark. All three patterns orbit the dynamic subject while keeping it on the optical axis. Each trajectory is parameterized by a per-frame (yaw, pitch, dolly) triple, interpolated linearly over the 49 frames of the clip:

- **f1** sweeps yaw from $-60^\circ$ to $+60^\circ$ at the source camera distance (pitch $0^\circ$, dolly $1.0$);
- **f2** sweeps yaw from $+30^\circ$ to $-30^\circ$ at a fixed pitch of $-10^\circ$ and $1.3\times$ the source camera distance;
- **f3** sweeps yaw from $+30^\circ$ to $-30^\circ$ at a fixed pitch of $+10^\circ$ while dollying in from $1.0\times$ to $0.7\times$ the source camera distance.

**Alignment conditioning.** Each pattern sweeps the camera through the source viewpoint. The mid-clip frames pass close to the source camera, and the generated video retains a large overlap with the source video. This overlap anchors the estimated trajectory to the source content and keeps the $\mathrm{Sim}(3)$ alignment of Appendix D well conditioned. DAVIS-TRAJ therefore does not exhibit the degenerate alignments removed by the filtering described there.

**Growing-magnitude trajectories.** For the generalization study of Sec. 4.2, we construct symmetric yaw sweeps on ten of the benchmark scenes, with per-side amplitudes from $10^\circ$ to $90^\circ$ (total sweep $20^\circ$–$180^\circ$) at zero pitch and unit dolly.

## D EVALUATION PROTOCOL

**Trajectory alignment.** For each generated clip, we run VGGT-Omega (Wang et al., 2026a) on the source and generated videos jointly, so their estimated cameras live in one coordinate frame. Given the estimated source cameras $\{(R_i^e, C_i^e)\}$ and their ground-truth counterparts $\{(R_i^g, C_i^g)\}$, we recover a global similarity transform $(s, R, \mathbf{t})$ that maps the estimated source trajectory to ground truth. The rotation $R$ is obtained by solving the orthogonal Procrustes problem on the camera orientations: we form the correlation matrix $M = \sum_i R_i^g (R_i^e)^\top$ and take its SVD $M = U\Sigma V^\top$; the optimal rotation is $R = UDV^\top$ where $D = \mathrm{diag}(1, 1, \det(UV^\top))$ guards against reflections. This orientation-based recovery is more robust than center-only SVD when the camera path is near-planar or short. The scale $s$ and translation $\mathbf{t}$ are then the least-squares fit of the camera centers given that rotation: $C^g \approx s\,R\,C^e + \mathbf{t}$. Since the generated cameras share the coordinate frame of the estimated source cameras, the same transform brings them into the ground-truth world, and we can compare them directly with the requested target trajectory, following the protocol described by Lin et al. (2026).

**Metrics.** We report three pose metrics, all computed per-frame and averaged over $T$ generated frames:

$$\text{RotErr} = \frac{1}{T}\sum_{i=1}^{T} \arccos\left(\frac{\mathrm{tr}(R_i^{\text{tgt}\top} R_i^{\text{gen}}) - 1}{2}\right), \tag{3}$$

$$\text{TransErr} = \frac{1}{\bar{d}}\sqrt{\frac{1}{T}\sum_{i=1}^{T} \|C_i^{\text{tgt}} - C_i^{\text{gen}}\|^2}, \tag{4}$$

$$\text{IntrErr} = \frac{1}{T}\sum_{i=1}^{T} \left|\text{FOV}_v(f_i^{\text{tgt}}) - \text{FOV}_v(f_i^{\text{gen}})\right|, \tag{5}$$

where $\bar{d}$ is the median source scene depth (normalizing by $\bar{d}$ removes the bias from each dataset's arbitrary world scale, enabling fair cross-dataset comparison), $\text{FOV}_v(f) = 2\arctan\!\left(\frac{1}{2f}\right)$ in degrees, and $f$ is the height-normalized focal length ($f_y/H$).

**Common clip set and degenerate alignment filtering.** All methods are evaluated on the maximum common clip set. We remove clips where RotErr $> 60^\circ$ in any method or seed, indicating a catastrophically wrong alignment rather than a failure of the generated video itself (we verified this through manual inspection). These failures occur exclusively on the Vista4D-Eval benchmark, where across all methods and seeds, 7/110 clips trigger this filter, typically when the source camera is nearly static and the overlap between source and target views is so small that, after source alignment, the target trajectory becomes mirrored relative to the ground truth. Some clips that escape filtering still suffer from elevated errors due to the same low-overlap pathology, though to a

Table 6: **Base models and inference time.** All methods run with 50 diffusion steps on a single NVIDIA A100 80 GB GPU and generate at their native resolution.

| Method | Base model | Output | Time (min) |
|---|---|---|---|
| ReCamMaster | Wan2.1-T2V-1.3B | 81×480×832 | 9.0 |
| TrajectoryCrafter | CogVideoX-Fun 5B | 49×384×672 | 2.7 |
| GEN3C | Cosmos-Predict1 7B | 121×704×1280 | 18.5 |
| Vista4D | Wan2.1-T2V-14B | 49×384×672 | 19.3 |
| OURS | Wan2.1-T2V-14B | 49×384×672 | 13.0 |

Table 7: **Camera control accuracy with Pi3 reconstruction.** *Left*: filtered (RotErr $\leq 60°$ in any method, maximum common clip set). *Right*: raw Vista4D-Eval (all 110 clips, no filtering). DAVIS-TRAJ has no filtered clips so only one column group is shown. 3-seed mean. **Bold** denotes the best and underline denotes the second best among generated methods (Point cloud render excluded from ranking and typeset in italics).

| Method | DAVIS-TRAJ (72 clips) | | | Vista4D-Eval (106 clips) | | | Vista4D-Eval raw (110 clips) | | |
|---|---|---|---|---|---|---|---|---|---|
| | RotErr ↓ | TransErr ↓ | IntrErr ↓ | RotErr ↓ | TransErr ↓ | IntrErr ↓ | RotErr ↓ | TransErr ↓ | IntrErr ↓ |
| ReCamMaster | 14.00 | 0.586 | 3.19 | 13.23 | 0.549 | 10.67 | 13.56 | 0.563 | 10.67 |
| TrajectoryCrafter | 1.79 | 0.256 | 1.52 | 6.62 | 0.500 | 7.55 | 7.71 | 0.504 | 7.57 |
| GEN3C | 1.88 | 0.267 | **1.41** | 4.05 | 0.438 | 7.52 | **4.25** | 0.433 | 7.52 |
| Vista4D | 2.90 | 0.304 | 2.06 | 4.07 | **0.382** | 7.23 | 4.67 | **0.388** | 7.26 |
| **OURS** | **1.56** | **0.247** | 1.59 | **3.82** | 0.419 | **6.96** | 4.39 | 0.427 | **6.97** |
| Point cloud render | *1.37* | *0.226* | *1.44* | *3.62* | *0.438* | *6.74* | *6.32* | *0.457* | *6.81* |

lesser degree. Our proposed benchmark (DAVIS-TRAJ) does not exhibit this issue (Appendix C). This filtering ensures that the reported numbers reflect genuine camera control quality rather than reconstruction failures of the pose estimator on pathological inputs, and the maximum-common-set constraint guarantees that all methods are scored on identical clips so no method benefits from a selective evaluation subset.

# E ADDITIONAL QUANTITATIVE RESULTS

## E.1 INFERENCE TIME

Table 6 compares the base models and per-clip inference time of all methods, measured on the same DAVIS-TRAJ clips with 50 diffusion steps on a single NVIDIA A100 80 GB GPU. The baselines differ in base model and output format, so absolute times are not directly comparable. Vista4D is the only baseline that uses the same base model as MANIFOLD4D. Compared with Vista4D, MANIFOLD4D is about $1.5\times$ faster and achieves better camera control.

## E.2 PI3 POSE ESTIMATION RESULTS

Table 7 reports camera control accuracy estimated by Pi3 (Wang et al., 2026b) instead of VGGT-Omega. The left half applies the RotErr $\leq 60°$ filter on the maximum common clip set; the right half reports raw metrics on all clips with no filtering. Since catastrophic alignment failures are estimator-dependent, this filter is recomputed under the Pi3 estimates and retains 106 of the 110 clips. The "Point cloud render" row is the point-cloud-rendered video (no generation), included as a geometric reference and excluded from ranking. Under the Pi3 estimate all methods show inflated errors, but MANIFOLD4D still ranks best overall.

## E.3 DETAILED YAW-SWEEP RESULTS

These tables supplement Fig. 4 of Sec. 4.2 with the full numerical results. Table 8 reports the rotation and translation errors of each method at every amplitude. Table 9 reports the corresponding VBench aesthetic and imaging quality scores.

Table 8: **Camera control accuracy under growing motion amplitude.** RotErr and TransErr at each amplitude, averaged over ten DAVIS-TRAJ scenes. The amplitude is the per-side yaw angle, from 10° to 90° (total sweep 20°–180°). The point cloud render is included as a geometric reference and excluded from ranking.

| | RotErr ↓ (deg) | | | | | | | | | TransErr ↓ (normalized) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Yaw | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 |
| ReCamMaster | 7.32 | 9.21 | 11.44 | 14.43 | 17.67 | 19.55 | 22.28 | 38.94 | 43.19 | 0.290 | 0.348 | 0.510 | 0.622 | 0.560 | 0.689 | 0.721 | 0.918 | 0.853 |
| TrajectoryCrafter | 0.41 | **0.50** | 0.82 | 1.39 | 1.61 | 2.79 | 2.93 | 3.85 | 7.15 | **0.028** | **0.035** | **0.060** | 0.600 | 0.101 | 0.896 | 0.539 | 0.367 | 0.415 |
| GEN3C | 0.98 | 1.08 | 1.23 | 2.01 | 1.91 | 2.44 | 2.92 | 12.58 | 6.49 | 0.038 | 0.066 | 0.080 | 0.557 | 0.344 | 0.401 | 0.496 | 1.011 | 0.651 |
| Vista4D | 0.58 | 0.96 | 1.78 | 2.19 | 2.45 | 2.45 | 4.53 | 6.30 | 6.17 | 0.029 | 0.050 | 0.091 | 0.103 | 0.163 | 0.146 | 0.206 | 0.303 | 0.302 |
| OURS | **0.40** | 0.55 | **0.79** | **1.03** | **1.25** | **1.35** | **1.95** | **2.46** | **2.77** | 0.034 | 0.044 | 0.066 | **0.079** | **0.085** | **0.110** | **0.139** | **0.167** | **0.200** |
| Point cloud render | *0.29* | *0.38* | *0.53* | *0.72* | *0.88* | *0.98* | *1.29* | *1.67* | *2.34* | *0.018* | *0.032* | *0.046* | *0.062* | *0.079* | *0.101* | *0.130* | *0.164* | *0.194* |

Table 9: **Visual quality under growing motion amplitude.** VBench aesthetic and imaging quality scores at each amplitude, averaged over ten DAVIS-TRAJ scenes.

| | Aesthetic quality ↑ | | | | | | | | | Imaging quality ↑ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Yaw | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 |
| ReCamMaster | 0.492 | 0.485 | 0.486 | 0.475 | 0.462 | 0.458 | 0.445 | 0.485 | 0.485 | 0.620 | 0.609 | 0.606 | 0.592 | 0.586 | 0.580 | 0.571 | 0.607 | 0.601 |
| TrajectoryCrafter | 0.492 | 0.481 | 0.476 | 0.472 | 0.459 | 0.451 | 0.451 | 0.449 | 0.449 | 0.628 | 0.620 | 0.611 | 0.601 | 0.590 | 0.578 | 0.568 | 0.556 | 0.554 |
| GEN3C | 0.470 | 0.455 | 0.449 | 0.437 | 0.427 | 0.418 | 0.414 | 0.415 | 0.429 | 0.552 | 0.550 | 0.542 | 0.527 | 0.512 | 0.499 | 0.489 | 0.436 | 0.485 |
| Vista4D | 0.510 | 0.506 | **0.510** | 0.506 | **0.507** | 0.494 | 0.493 | 0.490 | 0.495 | 0.692 | **0.689** | 0.676 | **0.677** | 0.660 | 0.657 | 0.654 | 0.650 | **0.654** |
| OURS | **0.520** | **0.513** | 0.510 | **0.507** | 0.502 | **0.503** | **0.500** | **0.502** | **0.499** | **0.692** | 0.682 | **0.677** | 0.672 | **0.665** | **0.662** | **0.661** | **0.652** | 0.653 |

Table 10: **Full visual quality and temporal stability metrics.** FID/FVD are computed against the corresponding source clips, CLIP-T is the frame-wise text-image similarity, and the remaining five columns are VBench scores ($\times 100$): subject consistency (SC), background consistency (BC), motion smoothness (MS), aesthetic quality (AQ) and imaging quality (IQ). Mean over 3 seeds.

| Method | FID ↓ | FVD ↓ | CLIP-T ↑ | SC ↑ | BC ↑ | MS ↑ | AQ ↑ | IQ ↑ |
|---|---|---|---|---|---|---|---|---|
| DAVIS-TRAJ | | | | | | | | |
| ReCamMaster | 65.55 | **1122.9** | 0.2392 | **88.333** | 89.807 | **97.932** | 48.099 | 57.204 |
| TrajectoryCrafter | 85.27 | 1571.6 | 0.2361 | 84.866 | 88.519 | 95.911 | 46.899 | 57.038 |
| GEN3C | 89.76 | 1463.9 | 0.2298 | 84.543 | **90.624** | 95.627 | 47.245 | 53.218 |
| Vista4D | **55.64** | 1208.2 | **0.2508** | 86.980 | 90.420 | 95.677 | 51.096 | 64.914 |
| OURS | 57.55 | 1222.7 | 0.2444 | 86.393 | 89.367 | 95.122 | **51.138** | **65.384** |
| *Vista4D-Eval* | | | | | | | | |
| ReCamMaster | **92.79** | **1181.1** | 0.2478 | **91.295** | **91.656** | **98.886** | 52.338 | 58.694 |
| TrajectoryCrafter | 116.31 | 1581.4 | 0.2317 | 85.335 | 88.032 | 97.672 | 47.579 | 63.300 |
| GEN3C | 113.43 | 1534.3 | 0.2246 | 85.984 | 90.804 | 97.958 | 48.852 | 62.001 |
| Vista4D | 98.12 | 1404.6 | **0.2534** | 87.948 | 91.338 | 97.505 | **55.012** | **70.933** |
| OURS | 97.37 | 1360.2 | 0.2498 | 86.948 | 88.649 | 97.511 | 53.757 | 70.509 |

### E.4 FULL QUALITY METRICS

Table 10 shows the full quality metrics on both benchmarks, where CLIP-T is the frame-wise CLIP (Radford et al., 2021) text-image similarity and the remaining columns are VBench (Huang et al., 2024) scores. ReCamMaster leads on FVD, SC and MS on both benchmarks, and also on FID and BC on Vista4D-Eval. We attribute this to its considerably weaker trajectory control. Our method is on par with Vista4D on these metrics.

## F ADDITIONAL QUALITATIVE RESULTS

### F.1 VISUALIZATION QUALITY

Fig. 7 shows qualitative results on the iPhone dataset. The red guide lines highlight 3D inconsistency: Vista4D exhibits mild misalignment on the dynamic subject, while GEN3C and TrajectoryCrafter show visibly worse quality on dynamic objects. ReCamMaster loses trajectory control entirely and deviates substantially from the ground truth.

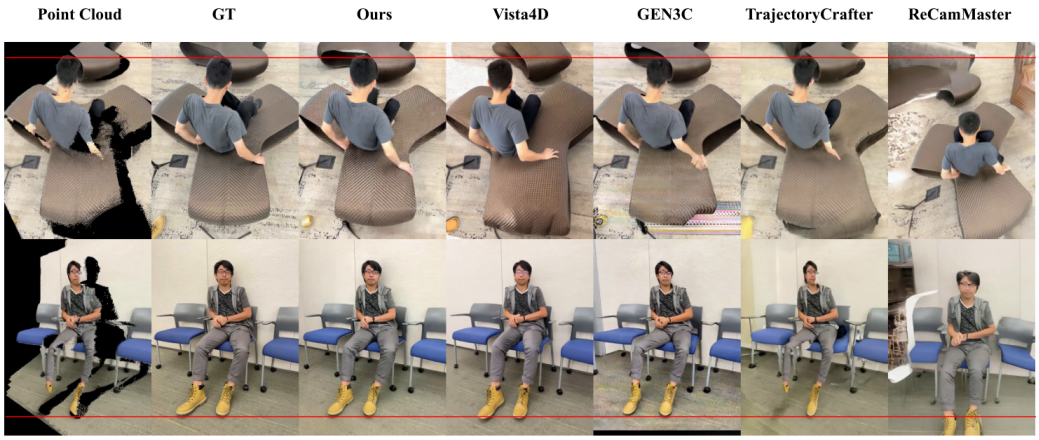


Figure 7: **Novel-view comparison on real iPhone multi-view captures.** Vista4D shows mild 3D inconsistency (red lines) and does not follow the point cloud colors; TrajectoryCrafter and GEN3C appear flat and miss fine human details; ReCamMaster lacks camera control.




Figure 8: **Robustness to incorrect dynamic geometry.** We omit the motion mask to simulate an erroneous point cloud: dynamic points from all frames are stacked together, smearing the rendered subject. MANIFOLD4D still recovers the correct motion from the source video.

### F.2 ROBUSTNESS TO IMPERFECT GEOMETRY

Fig. 8 visualizes the robustness test described in Sec. 4.3. When the motion mask is removed at render time, all dynamic points are stacked across frames, producing a render with incorrect dynamic geometry. Despite this corrupted input, MANIFOLD4D still recovers the correct motion from the source video. We do observe a moderate quality degradation in fine details, particularly a residual blur where the stacked points overlap, which is expected given the conflicting geometric signal.

### F.3 ABLATION STUDY ON ARCHITECTURE VARIANTS

Fig. 9 provides visualizations for the architecture ablation of Sec. 4.4. The Vista4D-style three-stream layout exhibits 3D inconsistency on dynamic objects, but the effect is milder than in the open-source Vista4D model. The GEN3C-style variant shows noticeable degradation in visual quality in some scenes.




Figure 9: **Qualitative ablation comparison.** The two architecture variants against the full model on Vista4D-Eval clips. The Vista4D-style three-stream layout exhibits 3D inconsistency on dynamic objects, while the GEN3C-style variant shows noticeable degradation in visual quality in some scenes. The full model preserves both geometric consistency and visual quality.

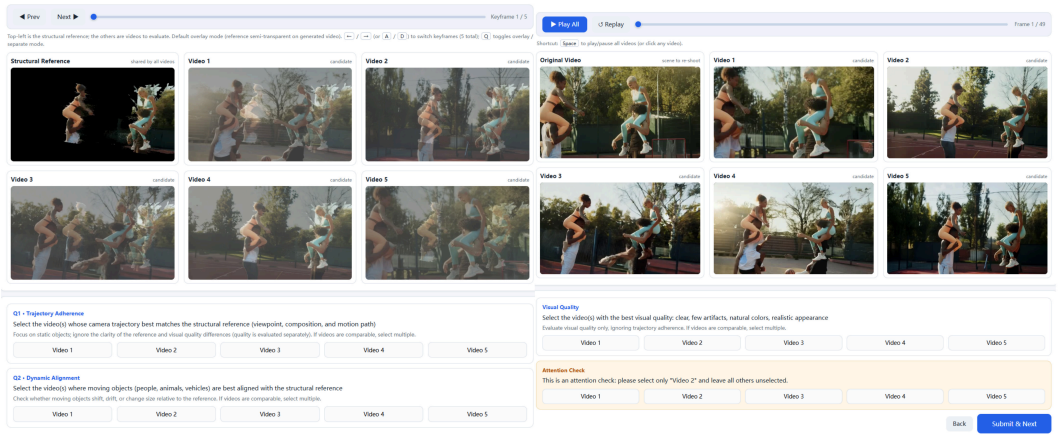

Figure 10: **User study evaluation interface. Left:** the point cloud render is overlaid on each generated video (toggleable with the Q key) for the trajectory-following and 3D-consistency questions. **Right:** the full generated video is played without overlay for the visual-quality question, presented separately to mitigate bias from the preceding overlay-based questions.

## G User Study Interface

Fig. 10 shows the user study interface. The left page evaluates trajectory adherence and 3D consistency of the dynamic subject; the right page evaluates visual quality. To assist evaluation, users can toggle an overlay of the point cloud render on the generated video. Five keyframes are provided for closer inspection. Visual quality is assessed on a separate page with the full video playback. This page is split from the trajectory and consistency evaluation to avoid bias. In our pilot study, users tended to pick the same video for all three criteria. Separating the pages substantially reduced this effect.