GENERATIVE CINEMATOGRAPHER: COMPOSING CAMERA AND OBJECT MOTION IN 3D

Anonymous authors
Paper under double-blind review

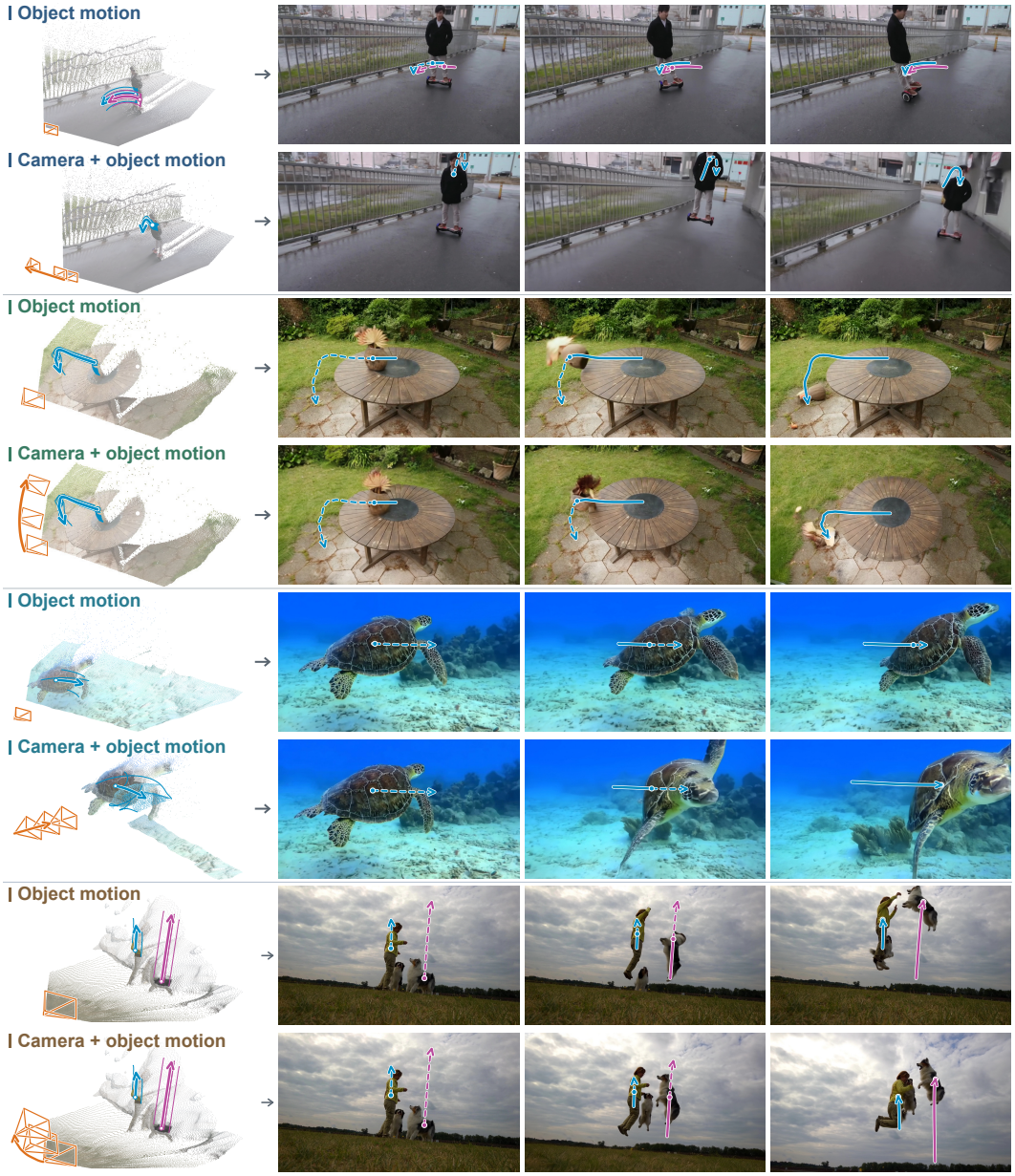


Figure 1: **Composing camera and object motion in 3D from a single image.** Artists choreograph foreground motion and camera paths in a shared 3D scene lifted from the input image. Each pair shows object motion with a fixed camera (top) and combined camera and object motion (bottom). The hoverboard rider follows a curved path or takes flight; the remaining pairs show a potted plant moving off a table, a swimming turtle, and a girl and dog jumping under separate controls. Each row shows the authored 3D controls followed by three generated frames. Orange denotes camera poses and paths; other colors denote foreground trajectories. Solid and dashed lines indicate elapsed and remaining motion, respectively. Best viewed as videos on the project website.

ABSTRACT

Existing controllable video generation methods can model camera motion or object motion, but most operate through image-space controls such as 2D trajectories or sparse drag signals. These controls are fundamentally ambiguous: the same image-plane motion may correspond to multiple underlying 3D motions, making camera motion and object motion difficult to disentangle under viewpoint changes. We present *Generative Cinematographer*, a controllable video generation framework that lifts a monocular image or video into an editable 4D scene scaffold where camera motion and foreground motion are authored jointly in a shared world frame. Our key observation is that world-space motion can be rasterized into dense correspondence maps that preserve persistent identities across viewpoints while remaining compatible with RGB-pretrained video diffusion models. The resulting representation supports sparse local 3D motion handles, coordinated camera–object choreography, and piecewise-rigid approximations of non-rigid motion without requiring physics simulators or category-specific priors. To preserve the generative prior of a pretrained video model, we freeze a Wan-based diffusion backbone and condition it through lightweight adaptation layers driven by the proposed correspondence representation. Experiments demonstrate coherent camera-relative motion, improved geometric consistency under viewpoint changes, and strong controllability across diverse real-world scenes.

1 INTRODUCTION

> *"Just as one can compose colors, or forms, so one can compose motions."*
> *— Alexander Calder (1933)*

Composing a cinematic shot means deciding how the camera and the subjects move together. Classical graphics systems such as the Virtual Cinematographer (He et al., 1996) compose shots by controlling a camera within a modeled 3D scene. We bring this world-space approach to video generation, allowing artists to jointly author camera and object motion using a 3D scaffold lifted from a single image (Fig. 1). Generative video models can now synthesize plausible motion from a single image, but giving artists precise control over these movements remains difficult. Text prompts cannot define the path of the camera exactly. Also, the 2D scribbles and drag hints have the problem that it is not clear what 3D motion was intended by those hints. For instance, a swipe gesture might imply camera motion or motion of an object on a static background.(Fig. 2). The resulting video may look plausible while following a different motion from the desired motion.

Many other controllable video methods specify foreground motion through pixel displacements (Wu et al., 2024; Geng et al., 2024), but cinematic scenes often involve local object deformations or occlusions that limit their generalizability. Approaches that use physics simulators or category-specific priors provide additional structure (Liu et al., 2024; Li et al., 2025b), but their applicability depends on the materials and objects they support. Learning control directly from video presents another challenge as reliable 3D motion annotations are scarce. Existing methods therefore often rely on synthetic training data, with a corresponding visual gap, or use per-pixel 2D motion as conditioning (Chu et al., 2025), leaving artists to specify motion in image space.

A natural way to express the intended motion is to place the camera and subjects in a shared 3D scene. Camera paths then describe how the viewpoint changes, while object trajectories describe how the scene itself changes. We follow this approach in **Generative Cinematographer**. Starting from a single image, we lift the scene into a colored point cloud. The artist selects a few foreground regions and specifies how each should move in 3D, along with a camera path. Each selected region acts as a *motion handle*: its points move together as a rigid group. Several handles can move different parts of a subject independently, providing a piecewise-rigid approximation to non-rigid motion without a skeleton, physics simulator, or category-specific prior.

To generate a video from these controls, we need to express them in a form that a pretrained video model can use. Recent methods already support joint camera and object control (Zheng et al., 2026; Zhang et al., 2026a). VerseCrafter, for example, renders a background point cloud and object-level Gaussian trajectories into RGB and depth maps. We instead pair a fixed identity for each local

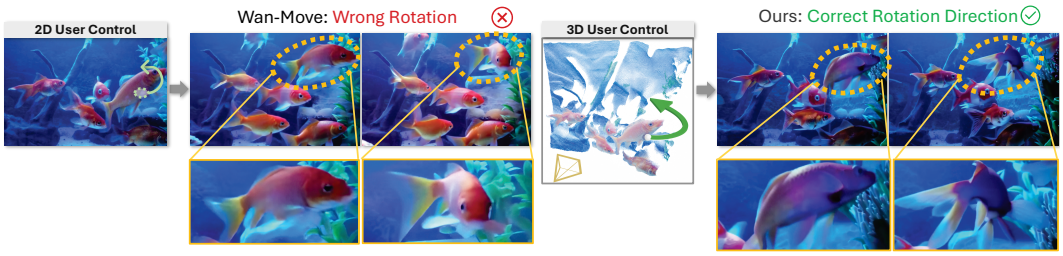


Figure 2: **Why author motion in 3D?** A 2D drag can leave the intended 3D rotation ambiguous. Here, the 2D-controlled baseline turns the fish in a direction inconsistent with the artist's intent, while our model follows the rotation specified in 3D. Enlarged views highlight the difference.

motion handle with the current world coordinates of its controlled points. The identity tells the model which region it is following; the coordinates specify how that region moves relative to the scene, even as the camera moves. We render this information into three maps: a background XYZ map for static world points, a foreground XYZ map for moving control points, and a foreground identity map that assigns each handle a fixed color. Both XYZ maps share the same world frame. Using local handles also lets the artist assign different motions to different parts of a subject.

We train the model using videos paired with controls recovered from their observed motion. For real videos, we estimate scene geometry and camera motion, then select small foreground regions and follow their points through the clip. These regions provide examples of the same motion handles that an artist would control. Synthetic videos provide ground-truth geometry and trajectories. We convert these controls into the three guidance maps and train a lightweight side branch and LoRA adapters to generate the observed video from its first frame and these maps. At inference, the artist supplies the camera path and handle trajectories in place of the recovered motion.

Our contributions are: (1) a **dense correspondence-map representation** that encodes point identity and position in a shared world coordinate system as rasterized maps compatible with pre-trained video VAEs; (2) a **local 3D motion-handle formulation** that approximates non-rigid motion through independently controlled local rigid regions, without requiring a simulator or category-specific prior; and (3) a **world-space authoring interface** that enables joint control of camera trajectories and local object motion from a single image.

2 Related Work

Virtual cinematography. The graphics community has long studied camera motion authoring. The Virtual Cinematographer (He et al., 1996) formalized shot composition as cinematographic idioms. Later work extended this to declarative camera planning (Christianson et al., 1996) and procedural camera control (Drucker et al., 1992). These systems assume a fully modeled 3D scene. In contrast, we use a 3D scaffold lifted from a single image to control the video generator.

Camera and motion control in video diffusion. MotionCtrl (Wang et al., 2024) and CameraCtrl (He et al., 2025a;b) condition generation on camera poses. GEN3C (Ren et al., 2025) renders a depth-derived 3D cache under target poses, while DimensionX (Sun et al., 2025) generates 3D and 4D scenes from one image. WorldReel (Fang et al., 2025) jointly generates RGB frames, point maps, camera trajectories, and flow for geometric and motion consistency. Generative Camera Dolly (Van Hoorick et al., 2024), TrajectoryCrafter (Yu et al., 2025), and FreeOrbit4D (Cao et al., 2026) redirect viewpoints in existing videos, with FreeOrbit4D reconstructing a geometry-complete 4D proxy. MotionCtrl also supports object control; CineMaster (Wang et al., 2025c) uses 3D box layouts, and Diffusion as Shader (Gu et al., 2025) uses 3D tracking videos for joint control.

Object motion control. DragNUWA (Yin et al., 2023), DragAnything (Wu et al., 2024), and ATI (Wang et al., 2025a) control motion through image-space trajectories; other methods use entities (Wan et al., 2024) or masks and boxes (Li et al., 2025a). Motion-I2V (Shi et al., 2024) predicts motion before synthesis, while Wan-Move (Chu et al., 2025) propagates latent features along trajectories. Motion Prompting (Geng et al., 2024) accepts sparse or dense trajectories, and Time-to-Move (Singer et al., 2025) guides a frozen generator with coarse animations. 3DTrajMas-

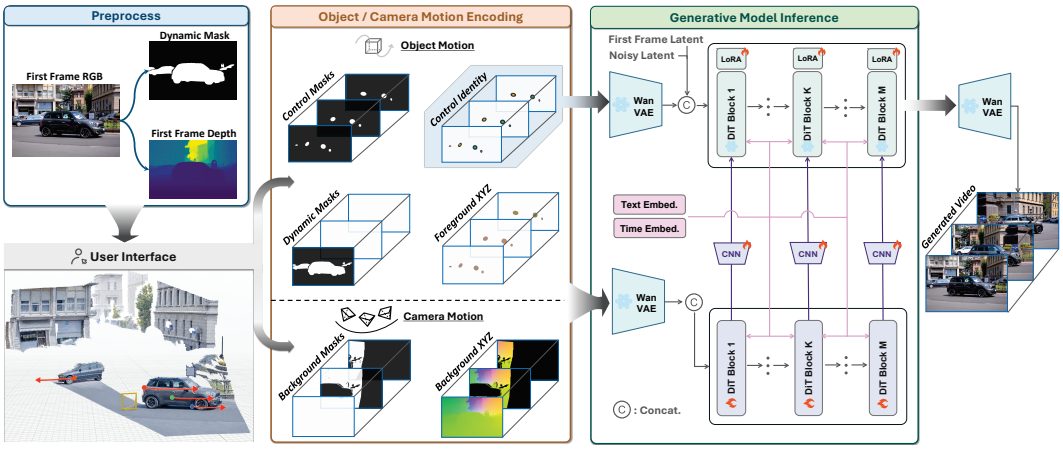


Figure 3: **Our 3D control workflow.** We estimate depth and a dynamic mask from an RGB image, then lift them into a 3D scene for motion authoring. The artist specifies camera and object motion in a shared world frame using a 3D interface such as Blender. We render these controls into background XYZ, foreground XYZ, and foreground identity maps, along with validity and dynamic masks (our main idea for obtaining control signals). A frozen Wan VAE encodes the guidance maps, which condition the main and side branches to generate a video following the authored motion.

ter (Fu et al., 2024) specifies entity-level 6DoF trajectories; MotionCanvas (Xing et al., 2025) maps scene-aware camera and object controls to screen-space conditions.

Joint camera and object control in 3D. SymphoMotion (Zhang et al., 2026a) combines geometry-aware camera guidance, 2D visual signals, and 3D trajectory embeddings; VerseCrafter (Zheng et al., 2026) renders a background point cloud and object Gaussian trajectories into conditioning maps. FlexTraj (Zhang et al., 2026b) encodes persistent track and segmentation identities in control videos, while Edit-by-Track (Lee et al., 2025) edits videos through paired source and target 3D tracks. Go-with-the-Track (Namekata et al., 2026) uses reference-anchored tracks for compositing, camera control, and object motion. MoRight (Liu et al., 2026) decouples camera and object motion through canonical static-view trajectories and cross-view attention. UniCaMo (Vu et al., 2026) constructs motion-consistent noise with occlusion-informed warping. Our artist-placed local 3D handles share the camera's world frame. We rasterize persistent handle identities and current world-space positions into conditioning maps, jointly encoding camera and object motion.

3 Method

We lift a single image into a shared 3D world where the artist authors camera trajectories and local foreground motion through *motion handles*. We rasterize these controls into image-plane *guidance maps* that encode world-space positions and correspondence for video generation (Fig. 3).

3.1 Scene Scaffold and Motion Authoring

Given an image $\mathbf{x}_0$, we estimate depth $\mathbf{D}_0$ using an off-the-shelf estimator (Huang et al., 2025) and take the initial camera frame as the world frame. We back-project each pixel $\mathbf{u} = (u, v)$ as $\mathbf{p}_\mathbf{u} = \mathbf{D}_0(\mathbf{u})\mathbf{K}_0^{-1}(u, v, 1)^\top$, where $\mathbf{K}_0$ is the intrinsic matrix. Attaching the image colors yields a point cloud $\mathbf{P}$, which a predicted dynamic mask $\mathbf{M}^{\mathrm{dyn}}$ partitions into static background $\mathbf{P}^{\mathrm{bg}}$ and potentially mobile foreground $\mathbf{P}^{\mathrm{fg}}$.

The artist selects foreground regions in a 3D viewport and assigns each a trajectory. Each handle moves its selected points as a rigid group. For N handles, let $\mathcal{I}_i$ be the fixed point-index set of handle i. Its world-space support at frame t is

$$\mathcal{H}_{t,i} = \{\mathbf{p}_{t,j} \mid j \in \mathcal{I}_i\}, \quad i = 1, \ldots, N, \quad t = 0, \ldots, T, \tag{1}$$

where $\mathbf{p}_{t,j}$ is the point's position under the authored motion. The artist also specifies camera intrinsics and world-to-camera poses $\{\mathbf{K}_t, \mathbf{R}_t, \mathbf{t}_t\}_{t=0}^{T}$ in the same world frame. Supplementary Sec. D provides scene-construction details.

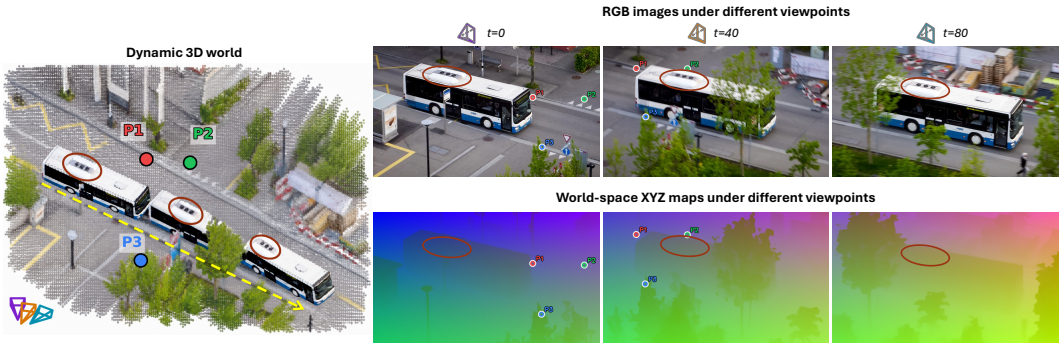


Figure 4: **World-space XYZ maps establish correspondences across viewpoints.** A moving bus is observed from different viewpoints at $t = 0$, $t = 40$, and $t = 80$. Static points **P1**, **P2**, and **P3** project to different image locations but retain their world-space coordinates and encoded colors. By contrast, the **highlighted region** on the moving bus changes its world-space position and therefore its XYZ color over time. We consequently introduce a separate, time-invariant foreground identity map to preserve correspondence, while the foreground XYZ map represents its time-varying position.

3.2 Motion Representation: Correspondence Maps and Rigid Handles

Our motion representation follows two principles: explicit correspondence and local rigidity. The first is motivated by motion correspondence in human visual perception (Ullman, 1979; Dawson, 1991); the second allows a few local controls to approximate complex foreground deformation (Sumner et al., 2007; Sorkine & Alexa, 2007).

- **Guidance correspondence maps.** We encode world-space XYZ coordinates as RGB values. Static background coordinates provide position and persistent identity. For moving foreground points, we separate current XYZ from fixed colors identifying handles across frames (Fig. 4).
- **Rigid motion handles.** Multiple independently controlled rigid handles approximate non-rigid motion—for example, moving a person's hands while keeping the torso still. This provides local control without authoring a dense deformation field or requiring an object template, skeleton, or category-specific motion prior.

Background and foreground XYZ maps ($\mathbf{F}_t^{\text{bg}}$, $\mathbf{F}_t^{\text{fg}}$). We project background and motion-handle points into each frame:

$$\mathbf{u}_t = \pi(\mathbf{K}_t(\mathbf{R}_t\mathbf{p}_t + \mathbf{t}_t)), \tag{2}$$

where $\pi((a, b, c)^\top) = (a/c, b/c)^\top$ denotes perspective division. Background points remain at $\mathbf{p}_t = \mathbf{p}_\mathbf{u}$, while handle points $\mathbf{p}_t \in \mathcal{H}_{t,i}$ follow their authored motion. We depth-test and rasterize these points, using $\mathbf{u}_t$ as the image location and normalized world-space XYZ as the RGB value.

Both maps share a per-axis normalization fixed throughout the clip. Let $\mathbf{p}_{\min}$ and $\mathbf{p}_{\max}$ be the coordinate-wise extrema of the valid first-frame points, with bounding-box center $\mathbf{o} = (\mathbf{p}_{\max} + \mathbf{p}_{\min})/2$ and half-range $\boldsymbol{\sigma} = (\mathbf{p}_{\max} - \mathbf{p}_{\min})/2$. We normalize world coordinates as

$$\mathcal{N}(\mathbf{p}) = \frac{1}{2}\left[\frac{\mathbf{p} - \mathbf{o}}{\boldsymbol{\sigma}} + \mathbf{1}\right], \qquad \text{where } \mathbf{1} = (1, 1, 1)^\top. \tag{3}$$

For points within the bounding box, $(\mathbf{p} - \mathbf{o})/\boldsymbol{\sigma}$ lies in $[-1, 1]^3$, and the shift and scaling map it to $[0, 1]^3$. The background and foreground RGB values are

$$\mathbf{c}_t^{\text{bg}} = \mathcal{N}(\mathbf{p}_\mathbf{u}), \qquad \mathbf{c}_t^{\text{fg}} = \mathcal{N}(\mathbf{p}_t), \quad \mathbf{p}_t \in \mathcal{H}_{t,i}. \tag{4}$$

A static background point keeps its world position and XYZ color as the camera moves; only its image location changes. The background map therefore indicates both *where* the point projects and *which* world point it represents. A foreground control point's XYZ color changes as it moves through the world. Using the same $(\mathbf{o}, \boldsymbol{\sigma})$ for both maps expresses foreground and background positions in a common coordinate frame and scale.

Foreground identity maps ($\mathbf{F}_t^{\text{id}}$). The foreground XYZ map alone cannot establish persistent correspondence because a moving point changes its world position and encoded color. We therefore assign each handle a time-invariant identity and render it separately.

For compact rasterization, we approximate $\mathcal{H}_{t,i}$ by a 3D Gaussian $\mathcal{G}_{t,i} = (\boldsymbol{\mu}_{t,i}, \boldsymbol{\Sigma}_{t,i})$, where $\boldsymbol{\mu}_{t,i}$ and $\boldsymbol{\Sigma}_{t,i}$ are the empirical mean and covariance of its points. Projection through Eq. 2 yields a 2D Gaussian with mean $\boldsymbol{\nu}_{t,i}$ and covariance $\boldsymbol{\Lambda}_{t,i}$. Let $\Omega_{t,i}$ denote its image-plane footprint.

When handle i is created, we sample a color $\mathbf{c}_i \in [0,1]^3$ and reuse it in every frame. Let $\alpha_{t,i}(\mathbf{u}_t) \in [0,1]$ be the Gaussian weight, largest at the projected center and decreasing toward the boundary. Within the footprint, we render

$$\mathbf{c}_t^{\mathrm{id}} = \alpha_{t,i}(\mathbf{u}_t)\mathbf{c}_i, \qquad \mathbf{u}_t \in \Omega_{t,i}, \tag{5}$$

Pixels outside all handle footprints are zero. Where footprints overlap, depth ordering selects the visible handle. The foreground XYZ and identity maps share this footprint, validity mask, and visible handle. The fixed base color $\mathbf{c}_i$ identifies the same handle across frames, while $\alpha_{t,i}$ determines its spatial weight. Rasterization produces three guidance maps $\mathbf{F}_t^{\mathrm{bg}}, \mathbf{F}_t^{\mathrm{fg}}, \mathbf{F}_t^{\mathrm{id}} \in [0,1]^{H\times W\times 3}$, with invalid pixels set to zero.

In summary, the maps provide complementary correspondence and motion cues. The background XYZ map supplies a static world reference for camera motion. For the foreground, the identity map identifies the projected handle, but does not explicitly encode its world-space motion. The foreground XYZ map supplies the control points' current world positions, allowing the model to interpret their motion relative to the background.

3.3 Architecture

We build on a pretrained Wan video diffusion model and freeze its VAE and base diffusion weights. We train only a lightweight side branch for geometric guidance, convolutional projections between branches, and LoRA adapters in the main branch.

The frozen VAE encodes all three guidance-map sequences. The main branch receives the noisy video latent, the reference-frame latent, and the encoded foreground identity sequence. The side branch receives all three encoded maps and binary masks marking valid background projections ($\mathbf{M}^{\mathrm{bg}}$), projected handle pixels ($\mathbf{M}^{\mathrm{fg}}$), and the first-frame dynamic region ($\mathbf{M}^{\mathrm{dyn}}$). It operates at half the main branch's spatial resolution and injects features through zero-initialized residual projections. Supplementary Sec. F details mask processing, input layouts, and feature transfer.

3.4 Sampling Motion Handles for Training

Training videos contain observed motion but no artist-authored handles. We use VIPE (Huang et al., 2025) to estimate depth, camera trajectories, and 3D point tracks from monocular clips. Taking the first frame as input, we use its dynamic mask to sample spatially separated handles from visible foreground points with valid geometry. Each handle groups nearby tracks within a scene-scaled 3D ball and follows their motion through the clip.

After filtering unreliable tracks and handles, we rasterize the accepted controls into foreground XYZ and identity maps. We vary the number and placement of handles across configurations to simulate partial artist control. Clips without valid foreground trajectories receive background-only conditioning. Supplementary Sec. G.1 provides the sampling and filtering details.

4 Experiments

Training Setup. We build upon Wan2.1-Fun-V1.1-14B-Control (Wan et al., 2025). We only train a small set of adaptation modules to learn the proposed geometry guidance maps. The model is trained for approximately 20k iterations on 4 H100 GPUs.

Datasets. We train on 40k video clips covering diverse camera and object motions. Approximately 5k clips come from synthetic datasets, including Dynamic Replica (Karaev et al., 2023) and PointOdyssey (Zheng et al., 2023), which provide ground-truth geometry and motion trajectories. The remaining clips come from real-world datasets, including SpatialVid-HQ (Wang et al., 2025b). For these clips, we estimate depth, camera poses, and dynamic masks using our geometry extraction pipeline and obtain motion trajectories with CoWTracker (Lai et al., 2026). All clips are processed and cropped to 480×832 resolution. We generate the missing video captions with Qwen2.5-VL-72B (Bai et al., 2025) and encode them with umT5 (Chung et al., 2023).

Table 1: Quantitative comparison on examples from Camera Shooting and Object Interactions. Metrics prefixed with R- are computed within the controlled foreground region. Subj., Bkg., and IQ denote VBench subject consistency, background consistency, and imaging quality, respectively.

	Camera Shooting									Object Interactions								
	Camera Traj.		Quality				Consistency			Camera Traj.		Quality				Consistency		
Method	Rot↓	Trans↓	PSNR↑	LPIPS↓	R-PSNR↑	R-LPIPS↓	Subj.↑	Bkg.↑	IQ↑	Rot↓	Trans↓	PSNR↑	LPIPS↓	R-PSNR↑	R-LPIPS↓	Subj.↑	Bkg.↑	IQ↑
Wan2.1	4.98	2.40	12.44	0.49	14.08	0.34	0.87	0.92	0.66	5.14	2.53	13.08	0.47	12.11	0.38	0.87	0.92	0.67
ATI	3.80	1.15	16.18	0.38	16.66	0.20	0.88	0.92	0.67	3.97	1.26	15.87	0.39	14.29	0.25	0.88	0.91	0.66
WanMove	3.73	1.02	16.58	0.36	17.13	0.19	0.90	0.93	0.69	3.76	1.08	16.51	0.37	**15.46**	0.22	0.89	**0.94**	0.67
VerseCrafter	3.92	1.27	15.27	0.41	15.82	0.27	0.90	0.92	**0.69**	4.02	1.24	13.94	0.43	13.27	0.33	**0.89**	0.92	**0.68**
SymphoMotion	5.04	2.45	12.29	0.50	13.94	0.35	0.86	0.91	0.65	5.20	2.58	12.91	0.48	11.96	0.39	0.86	0.92	0.66
Go-w-Track	**3.69**	1.07	16.61	**0.36**	17.11	0.19	**0.91**	**0.94**	0.69	**3.73**	1.08	**16.59**	0.37	15.43	0.22	0.88	0.92	0.67
Ours	3.70	**1.00**	**16.64**	0.36	**17.16**	**0.18**	0.90	**0.94**	0.68	3.74	**1.05**	16.54	**0.35**	15.40	**0.21**	0.89	0.92	0.67

Evaluation Setup. We evaluate on 50 examples from the DAVIS dataset and the SpatialVid test set, grouped by content into Camera Shooting and Object Interactions. Ground-truth videos and camera poses serve as references. We measure reference fidelity using full-frame PSNR and LPIPS (Zhang et al., 2018) and their foreground-region counterparts (R-PSNR and R-LPIPS). Following CameraCtrl (He et al., 2025a), we report camera rotation (Rot) and translation (Trans) errors. We assess video quality using the subject consistency, background consistency, and imaging quality metrics from VBench-I2V (Huang et al., 2024).

Baselines and Control Inputs. We compare with Wan2.1 (Wan et al., 2025) and five controllable video generation methods. In our evaluation, ATI (Wang et al., 2025a), WanMove (Chu et al., 2025), and Go-with-the-Track (Namekata et al., 2026) receive dense 2D foreground and background trajectories extracted from ground-truth videos. VerseCrafter (Zheng et al., 2026) and SymphoMotion (Zhang et al., 2026a) provide camera control and coarse object-level motion control. Our method uses sparse 3D motion handles and a target camera trajectory derived from the same reference motion. Dense ground-truth trajectories are unavailable when authoring new motion from a single image; a similar input asymmetry is discussed in concurrent work (Liu et al., 2026).

4.1 Main Results

Fig. 5 compares our method with Wan-Move, VerseCrafter, and Go-with-the-Track on artist-authored Living room, Bus, and Robotic arm sequences, showing the scene controls alongside temporally sampled outputs.

Tab. 1 reports the quantitative comparison on Camera Shooting and Object Interactions. Our method achieves the lowest translation error and R-LPIPS on both benchmarks and ranks second in rotation error behind Go-with-the-Track. On Camera Shooting, it also achieves the highest full-frame PSNR (16.64) and R-PSNR (17.16). On Object Interactions, it achieves the lowest full-frame LPIPS (0.35) and the second-highest PSNR (16.54), behind Go-with-the-Track (16.59).

The video quality and consistency metrics show complementary strengths. Go-with-the-Track achieves the highest subject consistency on Camera Shooting, while VerseCrafter achieves the highest imaging quality on Object Interactions. WanMove leads R-PSNR and background consistency on Object Interactions. Our method remains competitive across the video quality metrics, with its clearest advantages in translation accuracy and foreground perceptual fidelity.

Together, these results show that sparse 3D handles and camera trajectories support accurate control while the generative prior infers dense motion.

4.2 Ablation Study

We conduct ablation studies to evaluate the main design choices in our_method. We study the background XYZ map $\mathbf{F}^{\mathrm{bg}}$, the foreground XYZ map $\mathbf{F}^{\mathrm{fg}}$, and the side-branch resolution. All variants use the same training data, backbone, and optimization settings. Only the corresponding guidance component or architectural setting is changed. The results are reported on the Camera Shooting evaluation dataset in Tab. 2.

Removing $\mathbf{F}^{\mathrm{bg}}$ clearly reduces foreground control accuracy. Region LPIPS increases from 0.184 to 0.235, and Region PSNR drops from 17.16 to 16.20. This shows that background cor-

Table 2: Ablation study on 3D representation and different settings on Camera Shooting examples.

Method	R-LPIPS↓	R-PSNR↑	Subj. ↑	Bkg. ↑	IQ ↑
Ours (w/o $\mathbf{F}^{\mathrm{bg}}$)	0.235	16.20	0.889	0.937	0.683
Ours (w/o $\mathbf{F}^{\mathrm{fg}}$)	0.223	15.26	0.912	0.938	0.679
Ours ($s_{\mathrm{side}} = 1$)	0.189	17.17	0.897	0.942	0.686
Ours ($s_{\mathrm{side}} = 2$)	0.184	17.16	0.902	0.944	0.691

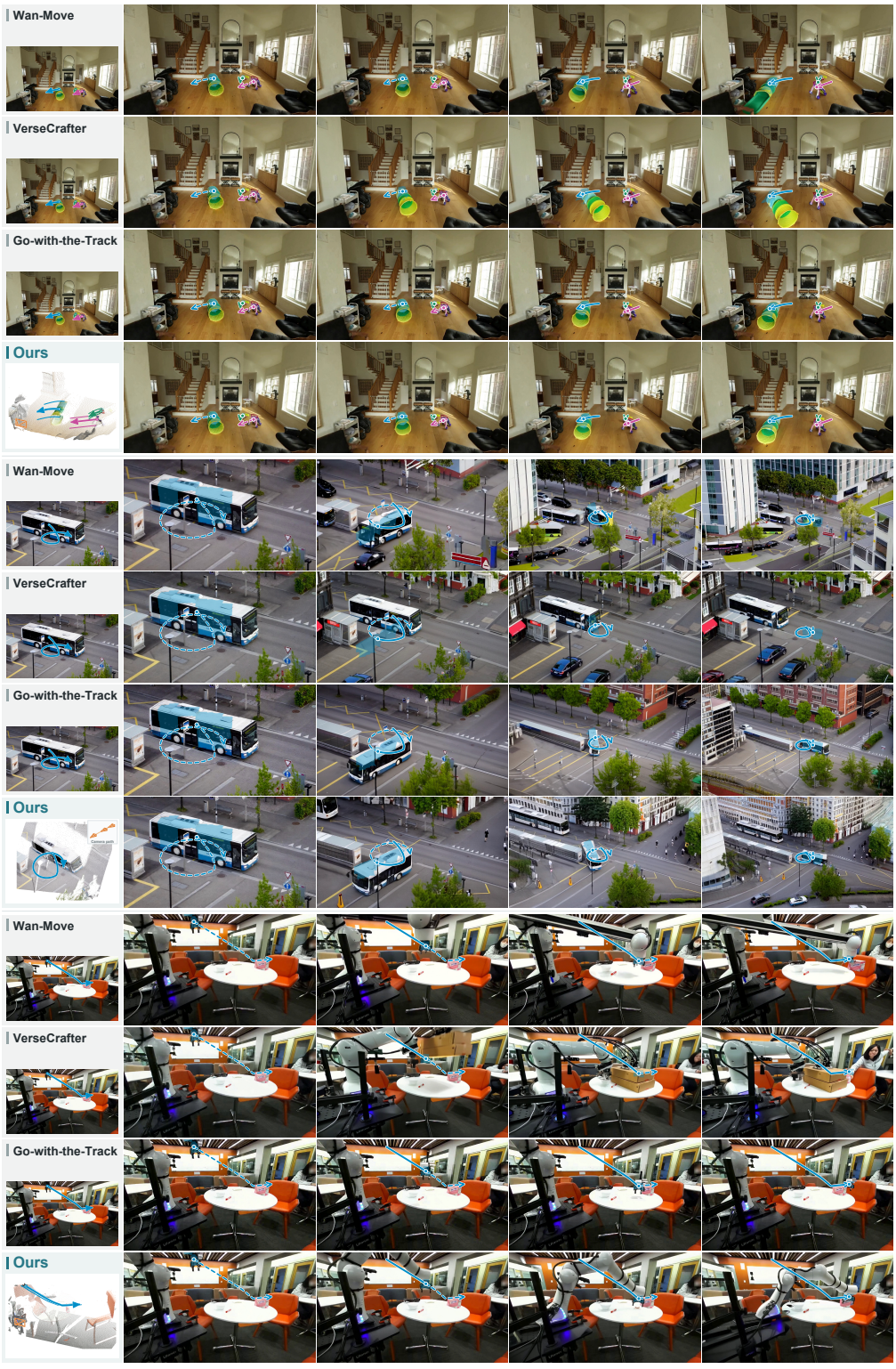


Figure 5: **Qualitative comparison on artist-authored motion.** Living room, Bus, and Robotic arm (top to bottom) each show Wan-Move, VerseCrafter, Go-with-the-Track, and ours, in row order. The first column shows 3D controls (ours) or shared 2D projections (baselines); four output frames sample each control interval. Solid/dashed paths show elapsed/remaining target motion, dots mark current targets, and translucent masks show projected target regions.



Figure 6: **Control map ablation.** The target is a fixed camera with the boat following the purple 3D trajectory. Foreground XYZ and identity maps alone (top) show background drift. Background XYZ alone (middle) provides no explicit object motion control. Full guidance (bottom) combines all three maps, better preserving the static background while guiding the boat.

respondence provides an important reference for separating camera motion from object motion.

As a qualitative illustration, Fig. 6 shows the intended task of moving the boat while keeping the camera fixed. The foreground-only result exhibits unwanted background motion, while the result with full foreground and background XYZ guidance better preserves the static scene. The middle row additionally shows background XYZ guidance only. The saved camera exports differ between the foreground-only and full-guidance runs, which limits attribution to background guidance alone.

Removing $\mathbf{F}^{\mathrm{fg}}$ also degrades the region-based metrics. Region LPIPS increases to 0.223, and Region PSNR drops to 15.26. This indicates that encoding foreground control points' world-space positions helps the model interpret their motion relative to the background. We further compare $s_{\mathrm{side}} = 1$ and $s_{\mathrm{side}} = 2$. The lower-resolution side branch with $s_{\mathrm{side}} = 2$ achieves similar region-level accuracy and better overall consistency and imaging quality. This suggests that low-resolution features are sufficient for the sparse geometry guidance maps.

Overall, the full model achieves the best balance between foreground control, temporal consistency, and visual quality. Our Figs. 1 and 9 illustrate the difficulty of achieving consistent 3D motion control using image-space trajectory conditioning. Because WanMove relies primarily on dense 2D tracks, multiple underlying 3D motions can project to similar image-plane trajectories, leading to ambiguous or geometrically inconsistent motion under viewpoint changes.

5 Discussion

We presented **Generative Cinematographer**, a system for jointly authoring camera and object motion in a shared 3D world lifted from a single image. Its guidance maps encode static background coordinates, persistent foreground handle identities, and current world-space positions in a form compatible with pretrained video diffusion models. This representation makes the intended 3D motion explicit where image-space trajectories alone leave it ambiguous. Our results show that a few local handles can compose camera and foreground motion, including a piecewise-rigid approximation to articulated motion, through lightweight adaptation of a pretrained video generator.

The single-image scaffold leaves occluded geometry unmodeled, requiring the generator to synthesize unseen surfaces under large viewpoint changes. Local rigid handles are better suited to articulated motion than to fracture, tearing, or fluid dynamics, and authoring many handles can become cumbersome for distributed deformation. Independently authored handles also impose no physical constraints or causal interactions between objects. Geometry completion, automatic handle placement, and learned interaction priors could extend the range of shots artists can compose.

REFERENCES

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, 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*, 2311.15127, 2023.

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.

David B. Christianson, Sean E. Anderson, Li-wei He, David H. Salesin, Daniel S. Weld, and Michael F. Cohen. Declarative camera control for automatic cinematography. In *Proceedings of the 13th National Conference on Artificial Intelligence (AAAI '96)*, pp. 148–155, 1996.

Ruihang Chu, Yefei He, Zhekai Chen, Shiwei Zhang, Xiaogang Xu, Bin Xia, Dingdong Wang, Hongwei Yi, Xihui Liu, Hengshuang Zhao, Yu Liu, Yingya Zhang, and Yujiu Yang. Wan-Move: Motion-controllable video generation via latent trajectory guidance. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2512.08765.

Hyung Won Chung, Xavier Garcia, Adam Roberts, Yi Tay, Orhan Firat, Sharan Narang, and Noah Constant. Unimax: Fairer and more effective language sampling for large-scale multilingual pretraining. In *ICLR*, 2023.

Michael R. W. Dawson. The how and why of what went where in apparent motion: Modeling solutions to the motion correspondence problem. *Psychological Review*, 98(4):569–603, 1991.

Steven M. Drucker, Tinsley A. Galyean, and David Zeltzer. CINEMA: A system for procedural camera movements. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics (I3D '92)*, volume 25, pp. 67–70. ACM, 1992.

Shaoheng Fang, Hanwen Jiang, Yunpeng Bai, Niloy J. Mitra, and Qixing Huang. WorldReel: 4D video generation with consistent geometry and motion modeling. *arXiv preprint arXiv:2512.07821*, 2025. URL https://arxiv.org/abs/2512.07821.

Xiao Fu, 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, 2024.

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. *arXiv preprint arXiv:2412.02700*, 2024.

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 *ACM SIGGRAPH 2025 Conference Papers*, 2025.

Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. CameraCtrl: Enabling camera control for text-to-video generation. In *International Conference on Learning Representations (ICLR)*, 2025a. arXiv:2404.02101.

Hao He, Ceyuan Yang, Shanchuan Lin, Yinghao Xu, Meng Wei, Liangke Gui, Qi Zhao, Gordon Wetzstein, Lu Jiang, and Hongsheng Li. Cameractrl ii: Dynamic scene exploration via camera-controlled video diffusion models. *ArXiv*, 2503.10592, 2025b.

Li-wei He, Michael F. Cohen, and David H. Salesin. The virtual cinematographer: A paradigm for automatic real-time camera control and directing. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '96)*, pp. 217–224. ACM Press, 1996. doi: 10.1145/237170.237259.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, volume 33, pp. 6840–6851, 2020.

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.

Jiahui Huang, Qunjie Zhou, Hesam Rabeti, Aleksandr Korovko, Huan Ling, Xuanchi Ren, Tianchang Shen, Jun Gao, Dmitry Slepichev, Chen-Hsuan Lin, Jiawei Ren, Kevin Xie, Joydeep Biswas, Laura Leal-Taixe, and Sanja Fidler. Vipe: Video pose engine for 3d geometric perception. *ArXiv*, 2508.10934, 2025.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *CVPR*, pp. 21807–21818, 2024.

Zeyinzi Jiang, Zhen Han, Chaojie Mao, Jingfeng Zhang, Yulin Pan, and Yu Liu. Vace: All-in-one video creation and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17191–17202, 2025.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. DynamicStereo: Consistent dynamic depth from stereo videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13229–13239, 2023.

Zihang Lai, Eldar Insafutdinov, Edgar Sucar, and Andrea Vedaldi. Cowtracker: Tracking by warping instead of correlation. *ArXiv*, 2602.04877, 2026.

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*, 2512.02015, 2025.

Quanhao Li, Zhen Xing, Rui Wang, Haidong Zhang, Qi Dai, and Zuxuan Wu. MagicMotion: Controllable video generation with dense-to-sparse trajectory guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025a.

Zhen Li, Hong-Xing Yu, Wei Liu, Yueqi Yang, Charles Herrmann, Gordon Wetzstein, and Jiajun Wu. WonderPlay: Dynamic 3d scene generation from a single image and actions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025b.

Shaowei Liu, Zhongzheng Ren, Saurabh Gupta, and Shenlong Wang. PhysGen: Rigid-body physics-grounded image-to-video generation. In *European Conference on Computer Vision (ECCV)*, pp. 360–378, 2024.

Shaowei Liu, Xuanchi Ren, Tianchang Shen, Huan Ling, Saurabh Gupta, Shenlong Wang, Sanja Fidler, and Jun Gao. MoRight: Motion control done right. *arXiv preprint arXiv:2604.07348*, 2026.

Koichi Namekata, Yash Kant, Zhizheng Liu, Ryan D. Burgert, Yuancheng Xu, Kuan Heng Lin, Emmett Steven, Julien Philip, Li Ma, Andrea Vedaldi, Paul Debevec, and Ning Yu. Go-with-the-Track: Video compositing and motion control with point tracking. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Papers*, SIGGRAPH Conference Papers '26. Association for Computing Machinery, 2026. doi: 10.1145/3799902.3811093. URL https://doi.org/10.1145/3799902.3811093.

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 (CVPR)*, pp. 6121–6132, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10684–10695, 2022.

Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, et al. Motion-I2V: Consistent and controllable image-to-video generation with explicit motion modeling. In *ACM SIGGRAPH 2024 Conference Papers*, pp. 1–11, 2024.

Assaf Singer, Noam Rotstein, Amir Mann, Ron Kimmel, and Or Litany. Time-to-move: Training-free motion controlled video generation via dual-clock denoising, 2025.

Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing (SGP)*, pp. 109–116. Eurographics Association, 2007.

Robert W. Sumner, Johannes Schmid, and Mark Pauly. Embedded deformation for shape manipulation. *ACM Transactions on Graphics*, 26(3):80, 2007. doi: 10.1145/1276377.1276478.

Wenqiang Sun, Shuo Chen, Fangfu Liu, Zilong Chen, Yueqi Duan, Jun Zhang, and Yikai Wang. DimensionX: Create any 3d and 4d scenes from a single image with controllable video diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

Shimon Ullman. *The Interpretation of Visual Motion*. MIT Press, 1979.

Basile Van Hoorick, Purva Tendulkar, Pulkit Suri, Kyle Park, Forrester Cole, Mark Sheinin, Roozbeh Mottaghi, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. *arXiv preprint arXiv:2405.14868*, 2024.

Long Vu, Tan Ngo, Animesh Karnewar, Amir Habibian, Binh-Son Hua, Hung Bui, Minh Hoai Nguyen, and Phong Nguyen-Ha. Track the noise, move the world: 3D-grounded motion-consistent noise for controllable video generation. *arXiv preprint arXiv:2607.02798*, 2026. URL https://arxiv.org/abs/2607.02798.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, Jianyuan Zeng, et al. Wan: Open and advanced large-scale video generative models. *ArXiv*, 2503.20314, 2025.

Zhang Wan, Sheng Tang, Jiawei Wei, Ruize Zhang, and Jianlong Liu. DragEntity: Trajectory guided video generation using entity and positional relationships. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024. doi: 10.1145/3664647.3681567.

Angtian Wang, Haibin Huang, Jacob Zhiyuan Fang, Yiding Yang, and Chongyang Ma. ATI: Any trajectory instruction for controllable video generation, 2025a.

Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Yi Zhang, Chang Zeng, Yanxi Zhou, Xiao-Xiao Long, Hao Zhu, Zhaoxiang Zhang, Xun Cao, and Yao Yao. SpatialVID: A large-scale video dataset with spatial annotations, 2025b.

Qinghe Wang, Yawen Luo, Xiaoyu Shi, Xu Jia, Huchuan Lu, Tianfan Xue, Xintao Wang, Pengfei Wan, Di Zhang, and Kun Gai. CineMaster: A 3d-aware and controllable framework for cinematic text-to-video generation, 2025c.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. MotionCtrl: A unified and flexible motion controller for video generation. In *ACM SIGGRAPH 2024 Conference Proceedings*, 2024. arXiv:2312.03641.

Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, David Junhao Zhang, Mike Zheng Shou, Yan Li, Tingting Gao, and Di Zhang. DragAnything: Motion control for anything using entity representation. In *European Conference on Computer Vision (ECCV)*, 2024. arXiv:2403.07420.

Jinbo Xing, Long Mai, Cusuh Ham, Jiahui Huang, Aniruddha Mahapatra, Chi-Wing Fu, Tien-Tsin Wong, and Feng Liu. MotionCanvas: Cinematic shot design with controllable image-to-video generation. In *ACM SIGGRAPH 2025 Conference Papers*, 2025. doi: 10.1145/3721238.3730604.

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*, 2308.08089, 2023.

Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. TrajectoryCrafter: Redirecting camera trajectory for monocular videos via diffusion models, 2025.

Guiyu Zhang, Yabo Chen, Xunzhi Xiang, Junchao Huang, Zhongyu Wang, and Li Jiang. SymphoMotion: Joint control of camera motion and object dynamics for coherent video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026a. URL https://arxiv.org/abs/2604.03723.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pp. 586–595, 2018.

Zhiyuan Zhang, Can Wang, Dongdong Chen, and Jing Liao. Flextraj: Image-to-video generation with flexible point trajectory control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026b.

Sixiao Zheng, Minghao Yin, Wenbo Hu, Xiaoyu Li, Ying Shan, and Yanwei Fu. VerseCrafter: Dynamic realistic video world model with 4D geometric control, 2026. URL https://arxiv.org/abs/2601.05138.

Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas. PointOdyssey: A large-scale synthetic dataset for long-term point tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

SUPPLEMENTARY MATERIAL

This supplement begins with our user study (Sec. A) and additional qualitative comparisons (Sec. B). It then reviews latent video diffusion (Sec. C) and details the scene scaffold (Sec. D), guidance maps (Sec. E), architecture (Sec. F), training (Sec. G), and inference (Sec. H). We conclude with failure cases and limitations (Sec. I).

A USER STUDY

Task and presentation. We conduct a blinded A/B preference study on a fixed gallery of artist-authored camera and object controls. Each trial pairs our output with one baseline output for the same source image and intended edit. The interface first presents the original image, a 3D control visualization, and the corresponding 2D trajectory annotation. Participants must play the control video before opening the side-by-side comparison. The generated videos are labeled only A and B, and both must be played before a preference can be submitted. Participants select the video they prefer overall for the requested edit, considering both adherence to the shown control and visual coherence over time. Fig. 7 shows the control-review and A/B selection stages for the same example.

Preference results. Tab. 3 reports descriptive, vote-weighted preference rates: the proportion of A/B choices favoring our output in each baseline comparison. Our method receives 56–80% preference across the five comparisons.

Table 3: **A/B user-study preferences.** Each entry is the percentage of choices favoring our method over the named baseline. Values above 50% indicate a larger share of preferences for ours. Results are descriptive aggregate rates.

Compared against	Ours preferred (%)
Wan-Move	72.0
ATI	80.0
Go-with-the-Track	56.0
VerseCrafter	60.0
SymphoMotion	80.0

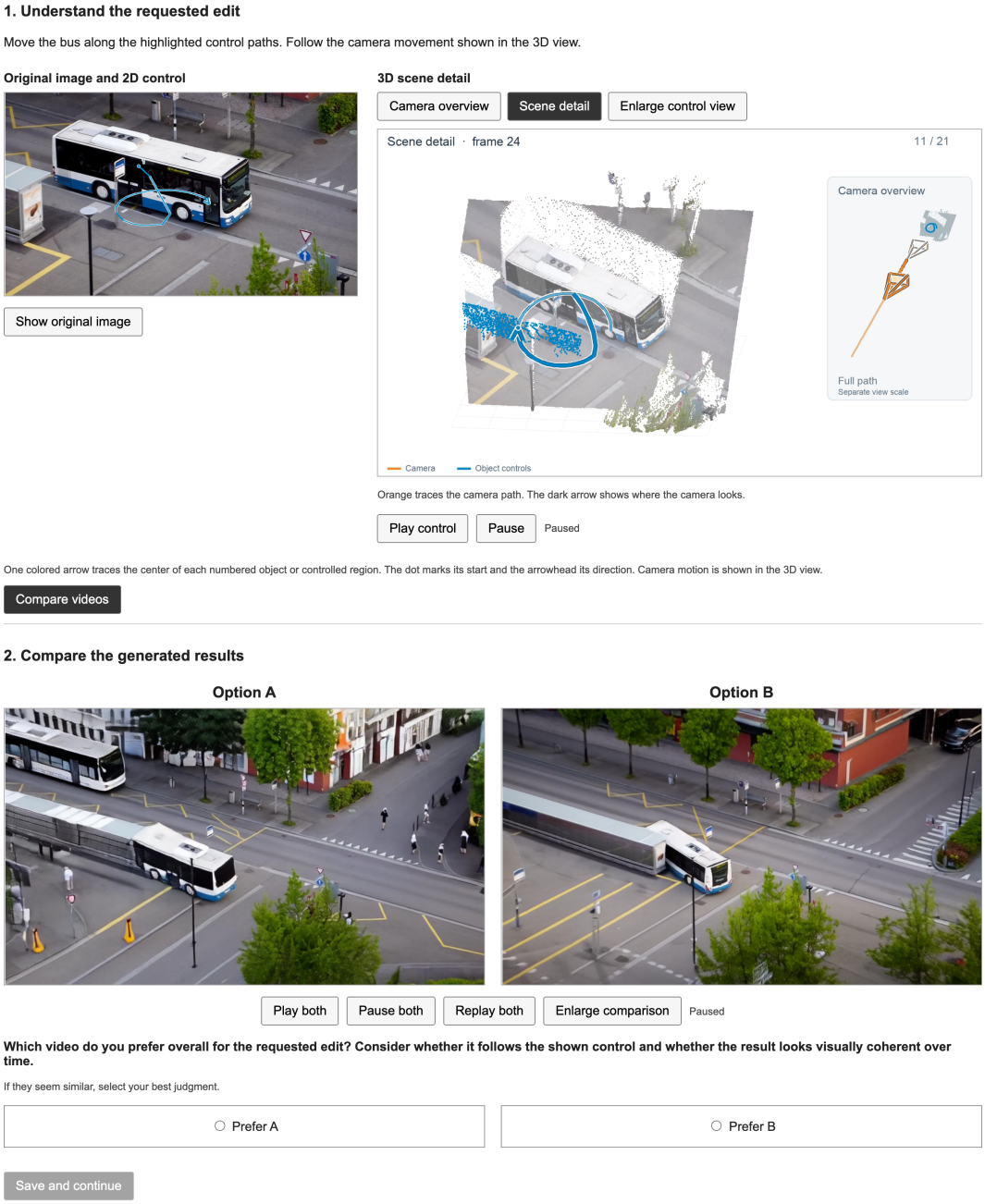


Figure 7: **User-study selection interface. Top:** Participants review the requested edit, the original image with its 2D control annotation, and the 3D scene with object and camera controls. **Bottom:** After watching the control video, they compare the corresponding generated outputs, labeled only Option A and Option B, using shared playback controls. Preference buttons become available after both outputs have been viewed. Participants select Prefer A or Prefer B and click Save and continue. Both screenshots show the same trial, with no preference selected.

B Additional Qualitative Comparisons

Camera and object motion. Fig. 8 compares outputs from ATI (Wang et al., 2025a), WanMove (Chu et al., 2025), and our method with ground-truth videos for the same foreground-control examples: hiking, camel, and dog sequences. Each scene contains four rows in this order, sampled at six aligned timesteps. The hiking and camel examples illustrate foreground displacement and changes in camera-relative appearance, while the dog sequence highlights articulated motion. Our outputs follow the specified foreground motion while preserving coherent scene context across the sampled frames. In the dog example, our generated poses more closely match the reference poses than the baseline outputs. These results complement the reconstruction metrics in the main paper.



Figure 8: Qualitative comparison across camera and object motion sequences. Each scene is sampled at six aligned timesteps for ATI (Wang et al., 2025a), WanMove (Chu et al., 2025), our method, and the corresponding ground-truth video. Our world-space representation better preserves camera-relative object motion, scene geometry, and foreground–background consistency over time, particularly under viewpoint changes and articulated motion (e.g., the dog sequence—object pose is closely aligned with the GT compared to other baselines).

Ambiguity in image-space control. Fig. 9 examines an edit that moves a teapot toward the camera while preserving its orientation in world space. Image-plane trajectories constrain projected motion but do not uniquely specify the underlying 3D transformation. In this example, WanMove produces geometrically inconsistent motion across the sampled frames. Our 3D handles specify the intended displacement in the canonical world frame, and the generated sequence better preserves the teapot's orientation and scene geometry. This example illustrates the benefit of making the intended 3D motion explicit when different spatial motions can have similar image-plane projections.

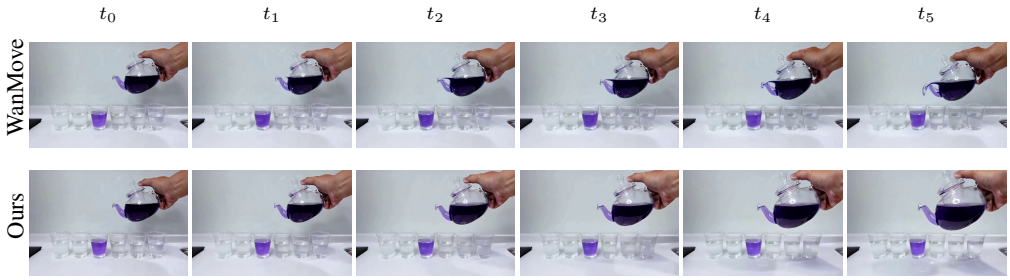


Figure 9: Comparison on an ambiguous motion-control example where the intended edit moves the teapot toward the camera while preserving its orientation in world space. Because the projected image-plane motion underconstrains the underlying 3D transformation, WanMove (Chu et al., 2025) produces geometrically inconsistent motion across frames. Our method resolves this ambiguity by authoring motion directly in a canonical world frame, yielding coherent camera-relative motion and stable scene geometry.

C Latent Video Diffusion Background

For readers less familiar with latent video diffusion, this section reviews the formulation that the main paper builds on. Video diffusion models (Ho et al., 2020; Blattmann et al., 2023; Wan et al., 2025) generate videos by learning to reverse a Gaussian noising process. To reduce compute, modern models denoise in a compressed latent space produced by a pretrained video VAE (Rombach et al., 2022; Blattmann et al., 2023).

Given an input video $\mathbf{V} \in \mathbb{R}^{(1+T)\times H\times W\times 3}$, where $1+T$ is the number of frames and $H \times W$ is the spatial resolution, the VAE encoder $\mathcal{E}$ compresses the video temporally by a factor f_t, spatially by a factor f_s, and expands the channel dimension to C:

$$\mathbf{z} = \mathcal{E}(\mathbf{V}) \in \mathbb{R}^{\left(1+\frac{T}{f_t}\right)\times \frac{H}{f_s}\times \frac{W}{f_s}\times C}. \tag{6}$$

The VAE decoder $\mathcal{D}$ reconstructs the video from the latent: $\hat{\mathbf{V}} = \mathcal{D}(\mathbf{z})$. The diffusion model operates entirely on $\mathbf{z}$. For Wan (Wan et al., 2025) specifically, $f_t = 4$, $f_s = 4$, and $C = 16$.

Two practical consequences of this formulation shape our design. First, conditioning inputs must look like RGB videos so that the same VAE can encode them; this constrains the form of our guidance maps. Second, the pretrained denoiser was trained on the latent distribution produced by this VAE, so substantially altering the input distribution (for instance, by joint RGB–geometry generation) tends to degrade the visual prior we wish to preserve.

D Scene Scaffold Details

Depth and camera estimation. The main paper estimates depth and the initial camera pose with VIPE (Huang et al., 2025). The intrinsic matrix $\mathbf{K}$ is taken from the depth pipeline's estimate when not provided as input. We do not apply additional hole filling: pixels with invalid depth are excluded from the point cloud and from all rasterized maps.

Dynamic mask. The dynamic mask $\mathbf{M}^{\mathrm{dyn}} \in \{0,1\}^{H\times W}$ is predicted at the first frame by GroundingSAM2. We use the predicted mask only as first-frame conditioning and pad the remaining frames with ones.

Coordinate convention. The initial camera frame is taken as the canonical world frame, with rotation $\mathbf{R}_0 = \mathbf{I}_3$ and translation $\mathbf{t}_0 = \mathbf{0}$. Subsequent camera poses are expressed as world-to-camera transforms. Per-frame intrinsics $\mathbf{K}_t$ are constant in our experiments unless the artist explicitly authors a zoom; we expose intrinsics per-frame in the construction (Sec. 3.2 of the main paper) for generality.

Back-projection and point cloud. Let $\Omega = \{0, \dots, W-1\} \times \{0, \dots, H-1\}$ denote the image domain and $\Omega_{\text{valid}} \subseteq \Omega$ the pixels with valid depth. For $\mathbf{u} = (u, v)$, let $\tilde{\mathbf{u}} = (u, v, 1)^\top$ be its homogeneous coordinate. Given the first-frame depth map $\mathbf{D}_0 \in \mathbb{R}^{H \times W}$, back-projection gives

$$\mathbf{p}_\mathbf{u} = \mathbf{D}_0(\mathbf{u})\mathbf{K}_0^{-1}\tilde{\mathbf{u}} = (x_\mathbf{u}, y_\mathbf{u}, z_\mathbf{u})^\top \in \mathbb{R}^3. \tag{7}$$

With the input RGB value $\mathbf{c}_\mathbf{u} = (R_\mathbf{u}, G_\mathbf{u}, B_\mathbf{u})^\top$, the colored point cloud is

$$\mathbf{P} = \big\{(\mathbf{p}_\mathbf{u}, \mathbf{c}_\mathbf{u}) \;\big|\; \mathbf{u} \in \Omega_{\text{valid}}\big\}. \tag{8}$$

The dynamic mask partitions $\mathbf{P}$ into the static background $\mathbf{P}^{\text{bg}}$ and potentially mobile foreground $\mathbf{P}^{\text{fg}}$. Throughout the construction, $\mathbf{u}$ denotes a 2D pixel coordinate, $\mathbf{p}$ a 3D world-space point, and $\mathbf{c}$ an RGB vector.

Authored handle support. Equation 1 defines the point set $\mathcal{H}_{t,i}$ using the fixed artist-selected indices $\mathcal{I}_i$. The selected points move as a rigid group under the handle's authored motion. Independently moving handles approximate non-rigid deformation, and additional handles allow finer local control.

E Guidance Map Construction Details

Shared coordinate normalization. The background and foreground XYZ maps use $\mathcal{N}$ from Eq. 3, with shared parameters derived from the first-frame point cloud:

$$\begin{aligned} \mathrm{RGB}_t(\mathbf{u}_t) &= \mathcal{N}(\mathbf{p}), \\ \mathbf{o} = \frac{\mathbf{p}_{\max} + \mathbf{p}_{\min}}{2}, &\qquad \boldsymbol{\sigma} = \frac{\mathbf{p}_{\max} - \mathbf{p}_{\min}}{2}, \end{aligned} \tag{9}$$

where $\mathbf{p}$ is the static world position for $\mathbf{F}^{\text{bg}}$ or the current handle position for $\mathbf{F}^{\text{fg}}$. The vectors $\mathbf{p}_{\min}$ and $\mathbf{p}_{\max}$ are the per-axis extrema over the valid first-frame points. Before computing the bounding box, we pull points beyond the 98th depth percentile back to that depth along their viewing rays. This prevents distant outliers from inflating $\boldsymbol{\sigma}$. At valid pixels, the inverse affine map is $\mathbf{p} = (2\,\mathrm{RGB} - \mathbf{1}) \odot \boldsymbol{\sigma} + \mathbf{o}$, where $\odot$ denotes element-wise multiplication. We compute $(\mathbf{o}, \boldsymbol{\sigma})$ once per clip and use it for both xyz maps in every frame. Invalid pixels are set to $\mathbf{0}$. Clips with extreme $\boldsymbol{\sigma}$ are excluded from training (see Sec. G).

Gaussian fitting for motion handles. We approximate each transformed point set $\mathcal{H}_{t,i}$ by a 3D Gaussian $\mathcal{G}_{t,i} = (\boldsymbol{\mu}_{t,i}, \boldsymbol{\Sigma}_{t,i})$ with empirical mean and covariance:

$$\boldsymbol{\mu}_{t,i} = \frac{1}{|\mathcal{H}_{t,i}|} \sum_{\mathbf{p} \in \mathcal{H}_{t,i}} \mathbf{p}, \qquad \boldsymbol{\Sigma}_{t,i} = \frac{1}{|\mathcal{H}_{t,i}|} \sum_{\mathbf{p} \in \mathcal{H}_{t,i}} (\mathbf{p} - \boldsymbol{\mu}_{t,i})(\mathbf{p} - \boldsymbol{\mu}_{t,i})^\top. \tag{10}$$

Degenerate handles — those with fewer than 5 points — are filtered out of the training set. A covariance is regularized by flooring its diagonal at a small fraction of the local point spacing, keeping the fit well-conditioned.

Fixed identity colors. When handle i is created, we sample an identity color $\mathbf{c}_i \in [0, 1]^3$ and keep it fixed throughout the video. Projecting $\mathcal{G}_{t,i}$ through Eq. 2 yields a 2D Gaussian with mean $\boldsymbol{\nu}_{t,i}$, covariance $\boldsymbol{\Lambda}_{t,i}$, and footprint $\Omega_{t,i}$. Within this footprint, the rendered value is $\alpha_{t,i}(\mathbf{u}_t)\mathbf{c}_i$ (Eq. 5), where $\alpha_{t,i} \in [0, 1]$ is largest at the center and decreases toward the boundary. Pixels outside all footprints are zero, and depth ordering selects the visible handle where footprints overlap. The foreground XYZ and identity maps share the same projected footprint, foreground validity mask, and visible handle. Fig. 10 shows examples of all three guidance maps.

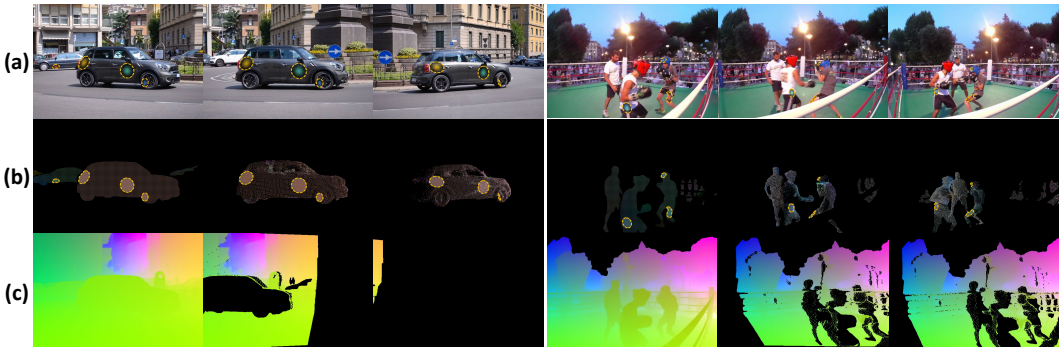


Figure 10: **Examples of the three guidance maps.** All maps use the same rasterized image-plane interface. **(a)** Foreground identity maps overlaid on RGB frames. Each artist-selected 3D handle region is rendered with a fixed identity color through its projected Gaussian footprint. **(b)** Foreground XYZ maps encode the current 3D positions of handle points as RGB values. **(c)** Background XYZ maps encode the world positions of valid static points. Both XYZ maps share one per-axis normalization, placing their values in the same coordinate system.

Validity masks. The construction in the main paper produces two pixel-wise validity masks alongside the three guidance maps. The background validity mask $\mathbf{M}^{\mathrm{bg}}$ marks pixels that receive at least one rasterized background point. The foreground validity mask $\mathbf{M}^{\mathrm{fg}}$ marks pixels covered by a valid projected handle footprint and is shared by $\mathbf{F}^{\mathrm{fg}}$ and $\mathbf{F}^{\mathrm{id}}$. The first-frame dynamic mask $\mathbf{M}^{\mathrm{dyn}}$ is passed in as a third cue, with temporal padding as described in Sec. D. Invalid guidance-map pixels are set to zero; the masks distinguish them from valid zero-valued coordinates.

Occupancy refinement. The background map is rasterized with a z-buffer and then morphologically refined. Closing only *proposes* hole pixels; a proposed hole becomes valid only when a region-local 3D Gaussian, fitted from the hole's immediate valid surface ring, actually yields a camera-ray intersection there. Unsupported holes are therefore left at zero and marked invalid, so the mask never claims geometry that was not reconstructed.

F Architecture Details

Backbone. We use Wan2.1-Fun-V1.1-14B-Control (Wan et al., 2025) as the latent video diffusion backbone. The backbone consists of 40 DiT blocks operating on the video latent produced by Wan's VAE. The VAE and base diffusion weights are frozen; only the side branch, inter-branch projections, and main-branch LoRA adapters are trained.

Side branch. The side branch follows the design of Wan-VACE (Jiang et al., 2025). It consists of 8 transformer blocks whose weights are initialized by copying from selected blocks of the main branch. The side branch contains approximately one-fifth of the main branch's parameter count.

The side branch operates at half the main branch's spatial resolution on the latent grid, yielding $4\times$ fewer spatial tokens. This reduces conditioning cost because the guidance maps are spatially sparse: most pixels in the foreground XYZ and identity maps are zero. The resolution ablation in the main paper evaluates this choice.

Projection layers. The two branches are bridged by a strided convolution and a transposed convolution operating on the latent grid. At the first side block, a strided $\mathrm{Conv3d}$ downsamples the main-branch features to the side resolution and adds them to the encoded guidance tokens; this layer is Dirac-initialized, so at the start of training it passes the main features through unchanged. Every side block then emits intermediate features that a $\mathrm{ConvTranspose3d}$ projects back to the main branch's resolution, added as residual hints to the corresponding main-branch block:

$$\mathbf{x}_{\ell+1} = \mathrm{Block}_{\ell}(\mathbf{x}_{\ell}, \mathbf{y}) + s_{\mathrm{vace}} \cdot h_{\ell}, \tag{11}$$

where $\mathbf{y}$ denotes the standard Wan conditioning inputs (text and time), h_ℓ is the projected side-branch hint, and s_{vace} is a scalar control strength. All output projections are zero-initialized, so $h_\ell = 0$ at step zero and the frozen backbone's behavior is reproduced exactly before any control is learned. The transposed convolution resolves its output padding per forward pass from the requested main-branch grid, so one set of weights serves any resolution.

LoRA adaptation. We additionally apply LoRA adapters (Hu et al., 2022) to the main-branch attention and feed-forward layers (q, k, v, o, and both feed-forward projections), with rank $r = 64$. LoRA gives the main branch a small amount of capacity to adjust to the new conditioning distribution without disturbing the bulk of the pretrained weights.

Input channel layouts. The main branch and side branch see different inputs.

Main branch. The main branch input concatenates the noisy video latent $\mathbf{z}_t$ with the VAE-encoded foreground identity map, and the encoded reference frame enters as an immutable one-frame token prefix rather than on the channel axis:

$$\text{main input} = \left[\mathbf{z}_t \,\|\, \mathcal{E}(\mathbf{F}^{\text{id}}) \,\|\, \mathbf{0}\right], \qquad \text{prefix} = \text{RefConv}(\mathbf{z}_0). \tag{12}$$

The zero block pads the control stream to the 48-channel input width of the Wan Control backbone.

Side branch. We reduce the binary masks to the VAE latent grid using max pooling, preserving small handle footprints during spatial and temporal reduction. Let $\widehat{\mathbf{M}}$ denote a processed mask tensor. The side branch concatenates the VAE-encoded guidance maps with these masks:

$$\begin{aligned}\text{side input} = \big[&\mathcal{E}(\mathbf{F}^{\text{bg}}) \,\|\, \widehat{\mathbf{M}}^{\text{bg}} \,\|\, \mathcal{E}(\mathbf{F}^{\text{fg}}) \,\|\, \widehat{\mathbf{M}}^{\text{dyn}} \,\|\\ &\mathcal{E}(\mathbf{F}^{\text{id}}) \,\|\, \widehat{\mathbf{M}}^{\text{fg}}\big].\end{aligned} \tag{13}$$

Each RGB guidance sequence contributes $C = 16$ VAE channels, and each processed mask contributes 4 channels in the input layout, for $16 + 4 + 16 + 4 + 16 + 4 = 60$ side-branch channels.

Text conditioning. Text prompts are encoded by umT5 (Chung et al., 2023) into text embeddings. The embeddings are injected into both the main branch and the side branch through Wan's standard text-conditioning interface; we make no modifications to the text pathway.

G Training Details

G.1 Synthetic Handle Sampling

VIPE (Huang et al., 2025) provides depth, camera trajectories, dynamic masks, and 3D point tracks for each monocular training clip. We use the first frame as input and construct synthetic controls from the observed foreground trajectories $\{\mathbf{p}_{t,j}\}_{t=0}^{T}$ in the canonical world frame. Let $\mathcal{J}^{\text{fg}}$ contain trajectories that have a finite 3D position in the first frame, are visible in that frame, and project inside $\mathbf{M}^{\text{dyn}}$. For each training variant, we sample the number of motion handles from a predefined range and select their seed trajectories from $\mathcal{J}^{\text{fg}}$.

Let $\mathbf{q}_{0,i} = \mathbf{p}_{0,j_i}$ be the initial center of motion handle i. We sample its radius relative to the scene scale s:

$$\begin{aligned}\widehat{r}_i &\sim \mathcal{U}(r_{\min}, r_{\max}),\\ r_i &= s\widehat{r}_i.\end{aligned} \tag{14}$$

The initial support contains all foreground trajectories within this radius:

$$\mathcal{I}_{0,i} = \left\{ j \in \mathcal{J}^{\text{fg}} \;\middle|\; \|\mathbf{p}_{0,j} - \mathbf{q}_{0,i}\|_2 \le r_i \right\}. \tag{15}$$

We reject a new center if it lies inside the support ball of an accepted motion handle. This encourages different motion handles to cover distinct object parts.

Unlike the fixed artist-selected support $\mathcal{I}_i$ in Eq. 1, synthetic support $\mathcal{I}_{t,i}$ can vary as tracks become invalid or reappear. We update each handle independently over time. At frame $t \ge 1$, we collect valid trajectories near the previous center $\mathbf{q}_{t-1,i}$ into a candidate set $\mathcal{C}_{t,i}$. We fit an empirical mean $\boldsymbol{\mu}_{t,i}$

and a regularized covariance $\mathbf{\Sigma}_{t,i}$ to these candidates. For each $j \in \mathcal{C}_{t,i}$, the squared Mahalanobis distance is

$$d^2_{t,i}(j) = (\mathbf{p}_{t,j} - \boldsymbol{\mu}_{t,i})^\top (\mathbf{\Sigma}_{t,i} + \epsilon \mathbf{I})^{-1} (\mathbf{p}_{t,j} - \boldsymbol{\mu}_{t,i}). \tag{16}$$

The refined support is $\mathcal{I}_{t,i} = \{j \in \mathcal{C}_{t,i} \mid d^2_{t,i}(j) \le \tau_{\mathrm{M}}\}$. If the seed trajectory remains in this set, its position defines $\mathbf{q}_{t,i}$. Otherwise, when the refined support is nonempty, we use its mean position. We then refresh the candidate set around the updated center, allowing a temporarily missing trajectory to rejoin the motion handle in a later frame.

A frame is valid when $|\mathcal{I}_{t,i}| \ge n_{\min}$. Otherwise, it contributes a temporal gap. Let $\mathcal{V}_i = \{t \mid |\mathcal{I}_{t,i}| \ge n_{\min}\}$ denote the valid frames. We retain a motion handle only when its valid-frame ratio satisfies

$$\frac{|\mathcal{V}_i|}{T+1} \ge \rho_{\min}. \tag{17}$$

We also measure the mean normalized footprint area of the candidate handle,

$$A_i = \frac{1}{|\mathcal{V}_i|} \sum_{t \in \mathcal{V}_i} \frac{|\Omega_{t,i}|}{HW}, \tag{18}$$

and require $A_i \in [A_{\min}, A_{\max}]$. This removes motion handles that are too small to be reliable or too large to represent local control. We resample highly overlapping motion handles until the desired number of distinct supports is obtained.

For every valid frame, we construct $\mathcal{H}_{t,i} = \{\mathbf{p}_{t,j} \mid j \in \mathcal{I}_{t,i}\}$ and rasterize it using the procedure in Sec. E. Invalid frames remain empty. We sample several handle configurations from each clip to vary the amount and location of foreground guidance. If $\mathcal{J}^{\mathrm{fg}}$ is empty, all foreground maps and masks are set to zero, producing a background-only training example.

Training objective. We adopt the flow-matching objective used by Wan:

$$\min_{\phi} \ \mathbb{E}_{(x,g),t} \left[\mathcal{L}_{\mathrm{FM}}(x, t; c(g), \phi) \right], \tag{19}$$

where x is a clean video latent, g is the precomputed geometry context, $c(g)$ is the conditioning tensor formed from g, t is the diffusion timestep, and ϕ collects the trainable parameters: the side branch, projection layers, and LoRA weights. All other parameters are frozen.

H Inference Details

Artist interface. At inference, an artist loads an image into our Blender GUI. The system runs the scaffold construction (depth, camera, dynamic mask, point cloud) once. The artist then interacts with the 3D viewport: handles are placed by clicking-and-dragging in the viewport, and the camera trajectory is authored by setting keyframes at desired frames or freely draw.

Partial guidance. At inference, the artist may provide only a few motion handles. Sampling multiple handle configurations during training (Sec. G.1) makes the model robust to different levels of foreground control. When no foreground handle is provided, the foreground maps are set to zero and the model relies on its learned motion prior and text condition. When no camera motion is authored, the camera remains at its initial pose.

I Failure Cases and Limitations

Depth failure. Monocular depth estimation can fail on scenes with strong reflections, thin structures (wires, hair), or aggressive transparency. When the lifted point cloud is geometrically wrong, the background correspondence map encodes an incorrect world structure, and the generated camera motion exhibits geometric inconsistencies during large viewpoint changes.

Piecewise-rigid approximation limits. Each handle translates rigidly. Complex non-rigid effects (fluid splash, fabric tear, fracture, deeply articulated motion) are well-approximated when handles are sufficiently dense and small. We do not currently expose deformation-mode handles (rotation, scaling, shear), which would extend the expressiveness of the representation but enlarge the authoring burden.

Motion beyond the normalized range. Both XYZ maps use the first-frame scene bounding box (Eq. 9). Points within this box map to $[0, 1]^3$, but a handle driven outside it can produce foreground XYZ values outside this range. Authoring within the scene extent keeps the guidance values in the RGB range; we expose the per-axis scale to the artist in the UI as a visual cue.