ObjCtrl-2.5D: Training-free Object Control with Camera Poses

Zhouxia Wang   Yushi Lan   Shangchen Zhou   Chen Change Loy
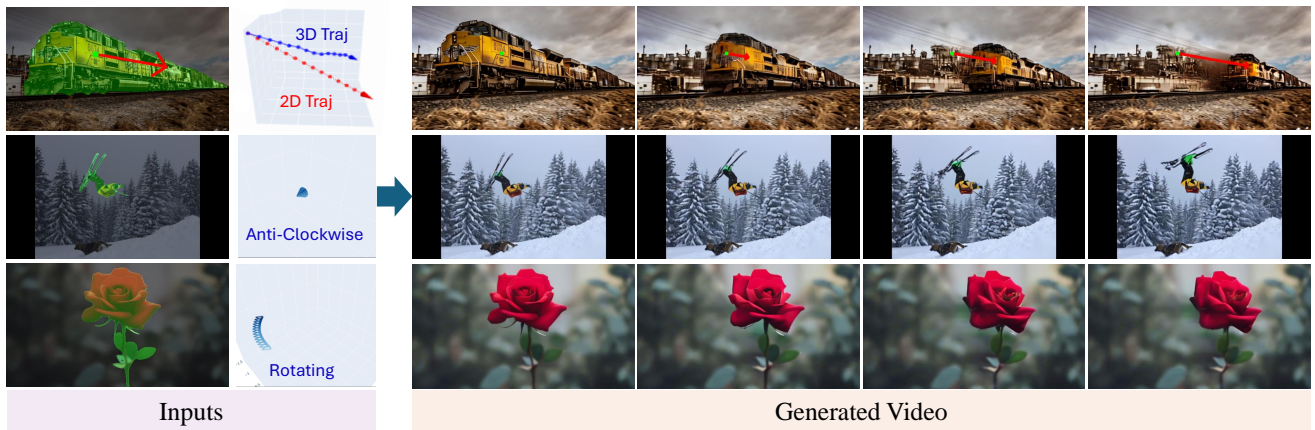
S-Lab, Nanyang Technological University

Figure 1. ObjCtrl-2.5D enables versatile object motion control for image-to-video generation. It accepts 2D trajectories, 3D trajectories, or camera poses as control guidance (all transformed to camera poses) and achieves precise motion control by utilizing an existing camera motion control module **without additional training**. *Unlike existing methods based on 2D trajectories, ObjCtrl-2.5D supports complex motion control beyond planar movement, such as object rotation, as demonstrated in the last row*. **We strongly recommend viewing the project page for dynamic results.**

Abstract

This study aims to achieve more precise and versatile object control in image-to-video (I2V) generation. Current methods typically represent the spatial movement of target objects with 2D trajectories, which often fail to capture user intention and frequently produce unnatural results. To enhance control, we present ObjCtrl-2.5D, a training-free object control approach that uses a 3D trajectory, extended from a 2D trajectory with depth information, as a control signal. By modeling object movement as camera movement, ObjCtrl-2.5D represents the 3D trajectory as a sequence of camera poses, enabling object motion control using an existing camera motion control I2V generation model (CMC-I2V) without training. To adapt the CMC-I2V model originally designed for global motion control to handle local object motion, we introduce a module to isolate the target object from the background, enabling independent local control. In addition, we devise an effective way to achieve more accurate object control by sharing low-frequency warped latent within the object's region across frames. Extensive experiments demonstrate that ObjCtrl-
2.5D significantly improves object control accuracy compared to training-free methods and offers more diverse control capabilities than training-based approaches using 2D trajectories, enabling complex effects like object rotation.

1. Introduction

Video generation seeks to produce high-quality videos from either a given text prompt (T2V generation) or a conditional image (I2V generation) and recently, numerous effective diffusion-based video generation models have emerged [1, 3, 5–7, 15–17, 24, 48, 61, 63, 65, 67]. The advancement of these models has spurred interest in developing more controllable generation, particularly for controlling the movement of objects within the generated video.

Most existing methods control objects using two-dimensional (2D) representations, such as bounding boxes [19, 23, 32, 49, 62] and trajectories composed of discrete points [22, 54, 58, 64]. These 2D guides specify only the spatial position of the moving object, while real-world objects move within a three-dimensional (3D) space. The lack of 3D information often results in unnatural video

1

outputs, as illustrated in Figure 2. The generated result in the first row is produced by DragAnything [58], a training-based object control method that relies on 2D trajectories as input. While the car relatively accurately follows the provided 2D trajectory, its movement is almost entirely horizontal toward the grass, which is unrealistic. In the reference video, the car moves not only toward the lower-left direction but also approaches the camera, as indicated by the decreasing depth along the 3D trajectory extracted from the reference video. We believe this depth information helps render the car to stay on the road instead of veering off into the grass in this sample.

To this end, we propose ObjCtrl-2.5D[1], a method that significantly enhances the accuracy of object motion control in T2V generation by explicitly leveraging 3D trajectories derived from 2D trajectories and scene depth information. Inspired by the effectiveness of camera motion control using camera poses in vision generation, such as MotionCtrl [54] and CameraCtrl [14], we propose to model the object movement with camera poses, which allows us to fully utilize the existing Camera Motion Control T2V (CMC-T2V) model for object motion control without any additional training.

Specifically, we first extend the 2D trajectory to 3D using the depth information extracted from the conditional image, and then project the 3D trajectory to camera poses via a triangulation algorithm [38, 39]. Given camera poses, existing CMC-T2V methods [14, 54] globally control camera motion in the generated videos, which conflicts with our need for local object motion control. To achieve training-free object motion control with the existing CMC-T2V models, we introduce a Layer Control Module (LCM) that isolates the target object from the background. This ensures that only the target object is influenced by the specified camera poses, while the background retains natural motion behavior. Additionally, we propose a Shared Warping Latent (SWL) to further improve object control accuracy by sharing low-frequency warping latents within the object's area in each frame, establishing an initial object movement that significantly influences the subsequent generation process. Leveraging the 3D information and a carefully designed object control model based on camera poses, ObjCtrl-2.5D achieves a significant improvement in control accuracy compared to previous training-free object control methods [19, 23, 32]. Furthermore, as ObjCtrl-2.5D can accept custom camera pose sequences, it enables more complex object motion control, such as object rotation, as illustrated in Figure 1.

In conclusion, this work makes the following main con-

---

[1]Our approach is termed 2.5D because, while combining a 2D trajectory with depth information produces a 3D trajectory that enables more realistic and controlled simulations of object movement in 3D space, it does not capture all aspects of 3D geometry.
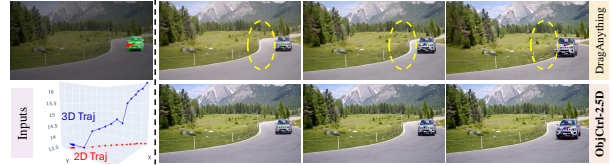

Figure 2. **Object control results using 2D and 3D trajectories.** On the right, the red line represents the 2D trajectory, the blue line indicates the 3D trajectory extracted from real-world video in DAVIS [31], and the green point marks the starting point of the trajectory. The training-based method DragAnything [58], which controls objects using a 2D trajectory, closely follows the specified path; however, it results in the car appearing to move horizontally toward the grass, which is atypical in real-world settings. By incorporating depth information from a 3D trajectory, our proposed method generates videos that not only follow the spatial trajectory but also achieve more realistic movement.

tributions: 1) ObjCtrl-2.5D extends 2D trajectories to 3D using depth information and represents these 3D signals with camera poses, achieving training-free object motion control with higher accuracy. 2) ObjCtrl-2.5D introduces a Layer Control Module and Shared Warping Latent, adapting the camera motion control module for effective object motion control and significantly enhancing object control performance. 3) ObjCtrl-2.5D achieves more complex and diverse object control capabilities compared to previous 2D-based methods.

2. Related Work

Video Generation. With the rising interest in content generation, video generation has become a prominent research area, producing a wealth of impactful work based on generative adversarial networks (GAN) [11, 28, 36, 37, 45, 47, 53] and diffusion models (DM) [1, 3, 5–7, 15–17, 24, 48, 61, 63, 65, 67]. Compared to GAN-based methods, diffusion models offer substantial advantages. To maximize the use of high-quality image datasets, most DM-based video generation models are derived from robust image-generation models, incorporating temporal modules and fine-tuning on video datasets. Notable examples include VDM [16], which builds upon a pixel-space diffusion model, and LVDM [15], which extends a latent diffusion model. Numerous models follow a similar framework, such as Align-Your-Latents [4], AnimateDiff [13], the VideoCrafter series [6, 7, 61], and SVD [3], among others. Furthermore, recent studies reveal that diffusion models based on transformers (DiT) [5, 17, 24, 63, 67] enhance both generation quality and scalability in video generation by replacing the conventional U-Net [35] backbone with a transformer architecture. This study adopts the U-Net-based diffusion model SVD [3], as it is relatively mature in video generation and includes various extensions, such
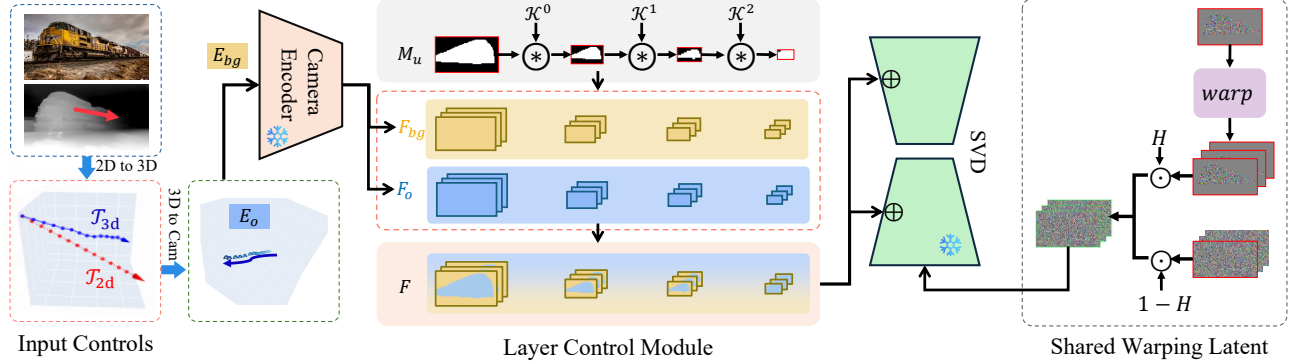
Figure 3. **Framework of ObjCtrl-2.5D.** ObjCtrl-2.5D first extends the provided 2D trajectory $\mathcal{T}_{2d}$ to a 3D trajectory $\mathcal{T}_{3d}$ using depth information from the conditioning image. This 3D trajectory is then transformed into a camera pose $\mathbf{E_o}$ via Algrithm 1. To achieve object motion control within a frozen camera motion control module, ObjCtrl-2.5D integrates a Layer Control Module (LCM) that separates the object and background with distinct camera poses ($\mathbf{E_o}$ and $\mathbf{E_{bg}}$). After extracting camera pose features via a Camera Encoder, LCM spatially combines these features using a series of scale-wise masks. Additionally, ObjCtrl-2.5D introduces a Shared Warping Latent (SWL) technique, which enhances control by sharing low-frequency initialized noise across frames within the warped areas of the object.

as control modules [14], which are valuable for exploring object control in this work. Besides, as an image-to-video generation model, SVD can tie the object and trajectories easily by drawing trajectory on the given conditional image.
Object Motion Control in Diffusion Video Models. Advances in basic video generation have improved developments in video customization, including motion control for both camera and object movement. Although previous works, such as Tune-A-Video [55], MotionDirector [66], LAMP [56], VideoComposer [52], and Control-A-Video [9], enable motion learning from specific reference videos or guided motion generation through depth maps, sketches, or motion vectors derived from reference videos, these approaches often lack user-friendliness. Given their flexibility and interactivity, trajectory [8, 12, 22, 25, 26, 42, 44, 54, 58, 64] and bounding box-based [19, 23, 32, 49, 62] methods have become popular in video motion control, generally classified as either training-based or training-free approaches. Training-based methods, including DragNUWA [64], DragAnything [58], and Image-Conductor [22], utilize trajectories to control both camera and object motion, while Boximator [49] achieves control using bounding boxes. MotionCtrl [54], by contrast, independently manages camera and object movements with separate camera and trajectory controls. Although effective, these methods demand significant computational resources for data curation and model training. Alternatively, training-free methods, SG-I2V [27] and [60] required per-sample optimization, and Direct-A-Video [62], PEEKA-BOO [19], TrailBlazer [23], and FreeTraj [32], enable object motion control by adjusting attention weights and initial noise according to specified trajectories and object bounding boxes. Although efficient and less computationally demanding, these methods are limited to 2D spatial object

movements and can only coarsely constrain generated models within the given bounding boxes, which limits accuracy and the ability to model diverse movements.

In contrast, ObjCtrl-2.5D presents a method for extending 2D trajectories into 3D, further modeling them through camera movement using a relatively precise transformation algorithm. By adapting a previous camera motion control module with delicate designs, this approach enables more accurate and versatile object motion control in image-to-video (I2V) generation, without additional training.

3. Methodology

3.1. Preliminary

Stable Video Diffusion (SVD). We adopt SVD [3], a publicly available and commonly used I2V diffusion model, as the basic model for our generation. SVD takes a conditional image $\mathbf{I_c}$ as input and generates a video with N frames $\{\mathbf{F}^0, \mathbf{F}^1, \ldots, \mathbf{F}^{N-1}\}$ using a conditional 3D U-Net [35] integrated with a latent denoising diffusion process [34].
CameraCtrl. Considering that object motion reflects the changes in spatial location across frames, we adopt CameraCtrl [14], a model that spatially represents camera poses using Plücker embeddings [43], as the basis for our object motion control. Generally, camera poses include intrinsic parameters, denoted $\mathbf{K} = [[f_x, 0, c_x], [0, f_y, c_y], [0, 0, 1]]$, and extrinsic parameters $\mathbf{E} = [\mathbf{R}|\mathbf{t}]$, where $\mathbf{R} \in \mathbb{R}^{3 \times 3}$ represents camera rotation and $\mathbf{t} \in \mathbb{R}^{3 \times 1}$ represents translation. Plücker embeddings enhance this representation by defining camera poses spatially as $\mathbf{p}_{x,y} = (\mathbf{o} \times \mathbf{d}_{x,y}, \mathbf{d}_{x,y}) \in \mathbb{R}^6$, where (x, y) indicates a position in image coordinates, $\mathbf{o} \in \mathbb{R}^3$ is equal to $\mathbf{t}$ and represents the camera center in world coordinates, and $\mathbf{d}_{x,y} \in \mathbb{R}^3$ is the direction vector from the camera center to pixel (x, y) in

world coordinates. Specifically,

$$\mathbf{d}_{x,y} = \mathbf{R}\mathbf{K^{-1}}[x, y, 1]^T + \mathbf{t}. \tag{1}$$

CameraCtrl extracts multi-scale camera motion information from the Plücker embeddings $\mathbf{P} \in \mathbb{R}^{N \times 6 \times H \times W}$, where N, H, and W represent the length, height, and width of the generated video, respectively, using a camera encoder. This camera motion information is then integrated into SVD, enabling global camera motion control.

3.2. ObjCtrl-2.5D

ObjCtrl-2.5D is a training-free model for object motion control, distinguishing itself from previous 2D-based approaches [19, 32, 58, 64] using 3D trajectories, which are attained by extending 2D trajectories with depth information. These 3D trajectories serve as control signals and are expressed as camera poses, allowing ObjCtrl-2.5D to leverage existing camera motion control models like CameraCtrl [14] for object motion control without additional training. Specifically, we first extend a 2D trajectory to 3D with depth from a conditional image. Subsequently, the 3D trajectory is modeled as a sequence of camera poses using triangulation [38, 39]. To adapt global motion methods, such as CameraCtrl, to local motion control, we introduce a Layer Control Module (LCM) that isolates the target object from the background, allowing for independent local manipulation. Additionally, Shared Warped Latents (SWL) is proposed to improve object control accuracy by sharing low-frequency warped latent information across the object area in each frame. Details of these components are provided in the following subsections.

3.2.1. 2D Trajectory to 3D to Camera Poses

2D Trajectory to 3D. The 2D trajectory is represented as $\mathcal{T}_{2d} = \{(x^0, y^0), (x^1, y^1), \dots, (x^{N-1}, y^{N-1})\}$, where $i \in [0, N-1]$. This trajectory is extended to 3D as $\mathcal{T}_{3d} = \{p^0, p^1, \dots, p^{N-1}\}$, with each point $p^i = (x^i, y^i, d^i)$ incorporating depth d^i, a value derived from the depth map $\mathbf{D_c}$ of the conditional image $\mathbf{I_c}$ using ZoeDepth [2]. Specifically, d^i is the depth value of $\mathbf{D_c}$ at the coordinate (x^i, y^i). To maintain smooth transitions, any abrupt depth changes between neighboring trajectory points are normalized. Additional details are provided in the supplementary materials.
3D Trajectory to Camera Poses. In this work, we transform the 3D trajectory to camera poses with triangulation algorithm [38, 39]. As illustrated in Figure 4, the object's movement from p^0 to p^i between frames $\mathbf{F}^0$ and $\mathbf{F}^i$ is modeled as a corresponding camera movement from $\mathbf{C}^0$ to $\mathbf{C}^i$, with all trajectory points mapped to the same point $\mathbf{P}_w = (x_w, y_w, z_w)$ in world coordinates. Since user-provided trajectories are often sparse, making it difficult to fully recover both rotation $\mathbf{R}$ and translation $\mathbf{t}$, we simplify by modeling the 3D trajectory as camera translation only,
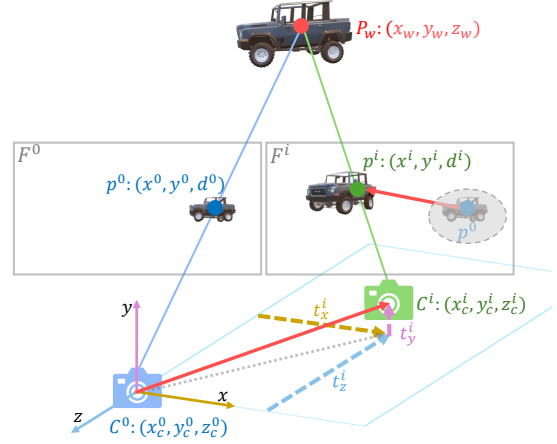

Figure 4. **3D Trajectory to Camera Poses.** We model the object movement in a video, indicated by a 3D trajectory, as the camera's location translation in 3D space. Details refer to Sec. 3.2.1 and Algorithm. 1.

omitting rotation. Thus, $\mathbf{R}$ is set as an identity matrix $\mathbf{I}$ for all camera poses, enabling us to represent the 3D trajectory with camera movement by solving for $\mathbf{t}^i = [t_x^i, t_y^i, t_z^i]$ using triangulation [38, 39].

Specifically, we first calculate the camera coordinates $\mathbf{C}^i = (x_c^i, y_c^i, z_c^i)$ for each frame, using the 3D trajectory points along with intrinsic parameters $\mathbf{K} = [[f_x, 0, c_x], [0, f_y, c_y], [0, 0, 1]]$:

$$x_c^i = z_c^i(x^i - c_x)/f_x; \quad y_c^i = z_c^i(y^i - c_y)/f_y; \quad z_c^i = d^i. \tag{2}$$

Following previous works [50, 59], $\mathbf{K}$ can be roughly estimated based on the spatial dimensions of the generated video or estimated with existing methods, such as UniDepth [30]. Then, we compute $\mathbf{P}_w = (x_w, y_w, z_w)$ with world-to-camera transformation, *i.e.*, $\mathbf{C}^i = [\mathbf{I}|\mathbf{t}^i][x_w, y_w, z_w, 1]^T$, attained:

$$x_w = x_c^i - t_x^i; \quad y_w = y_c^i - t_y^i; \quad z_w = z_c^i - t_z^i. \tag{3}$$

Drawing inspiration from DUSt3R [51], we set the first frame $\mathbf{F}^0$ as the canonical camera space, *i.e.*, $\mathbf{t}^0 = [0, 0, 0]$ and the subsequent frames are expressed in the same coordinate space as $\mathbf{F}^0$. Thus, $\mathbf{P}_w = (x_c^0, y_c^0, z_c^0)$ and:

$$t_x^i = x_c^i - x_c^0; \quad t_y^i = y_c^i - y_c^0; \quad t_z^i = z_c^i - z_c^0. \tag{4}$$

The pseudocode is given in Algorithm 1.

Note that while ObjCtrl-2.5D models the 3D trajectory as camera poses without rotation, it can also accept user-provided camera poses with rotation, thereby supporting motion control beyond mere translational movement in 3D space, such as object self-rotation (see last row in Figure 1)

4

3.2.2. Layer Control Module

To adapt CameraCtrl [14], originally designed for global motion control, to object-specific motion, we introduce Layer Control Module (LCM). This module separates the conditional image $\mathbf{I_c}$ into foreground and background layers using an object mask $\mathbf{M}_c$ generated via instance segmentation, such as SAM [21, 33]. The foreground layer is controlled by object-specific camera poses $\mathbf{E_o}$, derived from 3D trajectories outlined in Sec. 3.2.1, while the background layer is guided by background-specific poses $\mathbf{E_{bg}}$. These background poses can be customized, with options like $[\mathbf{I}|\mathbf{0}]$ allowing for a static background.

To extract camera features, $\mathbf{E_o}$ and $\mathbf{E_{bg}}$ are fed into the Camera Encoder, yielding $\mathbf{F_o} = \{f_o^0, f_o^1, \ldots, f_o^{S-1}\}$ and $\mathbf{F_{bg}} = \{f_{bg}^0, f_{bg}^1, \ldots, f_{bg}^{S-1}\}$, where S is the number of scales. These features are then fused with mask $\mathbf{M_o}$, which indicates the dominated area of $\mathbf{E_o}$, while $(1 - \mathbf{M_o})$ indicates the dominated area of $\mathbf{E_{bg}}$. To ensure $\mathbf{E_o}$ comprehensively covers the areas of the moving object across all the frames, we first attain the frame-wise object area $\mathbf{M_w} = \{m_w^0, m_w^1, \ldots, m_w^{N-1}\}$ from $\mathbf{M}_c$ using a geometric warping function $\mathrm{warp}(\cdot)$ [10, 18, 29, 41], where:

$$m_w^i = \mathrm{warp}(\mathbf{M}^0; \mathbf{D_c}, \mathbf{E_o}^0, \mathbf{E_o}^i, \mathbf{K}), \quad i \in [0, N-1], \tag{5}$$

where $\mathbf{D_c}$ is the depth, $\mathbf{E_o}^i$ is the object's camera pose for frame i, and $\mathbf{K}$ represents the intrinsic parameters. The union of these masks, $\mathbf{M_u} = \bigcup_{i=0}^{N-1} m_w^i$, defines the complete object area dominated by $\mathbf{E_o}$.

To prevent $\mathbf{M_u}$ from losing effectiveness during smaller-scale feature fusion, particularly for smaller target objects, we progressively dilate $\mathbf{M_u}$ at each scale using kernel $\mathcal{K}$. This process generates a set of dilated masks $\mathbf{M_o} = \{m_o^0, m_o^1, \ldots, m_o^{S-1}\}$, where

$$m_o^s = m_o^{s-1} * \mathcal{K}^{s-1}, \quad s \in [0, S-1], \quad m_o^{-1} = \mathbf{M_u}. \tag{6}$$

Then fused feature $\mathbf{F} = \{f^0, f^1, \ldots, f^{S-1}\}$ is:

$$f^s = f_o^s \odot \mathbf{m_o}^s + f_{bg}^s \odot (1 - \mathbf{m_o}^s), \quad s \in [0, S-1], \tag{7}$$

which is scale-wisely injected into SVD to control object motion in the generated video.

3.2.3. Shared Warping Latent

As a training-free approach, ObjCtrl-2.5D with LCM achieves good performance in object motion control compared to related methods. To further enhance control accuracy on challenging cases, such as generating uncommon object movements like a reversing boat (as shown in Figure 7), we introduce frame-wise shared low-frequency latents [32], *i.e.*, Shared Warping Latent (SWL). Unlike FreeTraj [32], which simply copies object latents, bounding with a box, from the first frame to all frames, we employ a geometric warping function $\mathrm{warp}(\cdot)$ [10, 18, 29, 41] to warp

---

Algorithm 1 Pseudocode of 3D Trajectory to Camera Poses.

```
def Traj3D_to_CameraPoses(T3d, fx, fy, cx, cy):
    '''
    Input:
        T3d: numpy.array, [N, 3], [frame_id, (x, y, d)]
        fx, fy, cx, cy: float, intrinsic paramters.
    Output:
        t: [tx, ty, tz]
    '''
    zc = T3d[:, 2]
    xc = (T3d[:, 0] - cx) * zc / fx
    yc = (T3d[:, 1] - cy) * zc / fy

    xw, yw, zw = xc[0], yc[0], zc[0]
    tx, ty, tz = xc - xw, yc - yw, zc - zw

    return [tx, ty, tz]
```

---

shared latent across frames, enabling a more precise object moving control.

Similar to Eq. 5, given the initial noise $\mathbf{z}$ of all the frame, we create a sequence of warped noise maps, $\mathbf{z}_w = \{\mathbf{z}_w^0, \mathbf{z}_w^1, \ldots, \mathbf{z}_w^{N-1}\}$ from $\mathbf{z}^0$, the first noise map in $\mathbf{z}$, as follows:

$$\mathbf{z}_w^i = \mathrm{warp}(\mathbf{z}^0; \mathbf{D_c}, \mathbf{E_o}^0, \mathbf{E_o}^i, \mathbf{K}), \quad i \in [0, N-1]. \tag{8}$$

To ensure that only latents within the object regions are shared across frames while preserving randomness in the background, we apply warping masks $\mathbf{M_w}$ to the warped noise, blending them back into $\mathbf{z}$ to produce $\mathbf{z_L}$:

$$\mathbf{z_L} = \mathbf{M_w} \odot \mathbf{z_w} + (1 - \mathbf{M_w}) \odot \mathbf{z}. \tag{9}$$

To mitigate the quality decrease of the generated video, only low-frequency information from $\mathbf{z_L}$ is retained:

$$\hat{\mathbf{z}} = \mathcal{FFT}_{3D}(\mathbf{z_L}) \odot \mathcal{H} + \mathcal{FFT}_{3D}(\mathbf{z}) \odot (1 - \mathcal{H}), \tag{10}$$

where $\mathcal{FFT}_{3D}$ denotes the 3D Fast Fourier Transform [57], $\mathcal{H}$ is a 3D low-pass filter, and $\hat{\mathbf{z}}$ serves as the noise at the T_{th} step in SVD.

4. Experiments

Experimental Settings. ObjCtrl-2.5D employs CameraCtrl [14], deployed on SVD [3], as the foundational image-to-video generation model. It supports diverse object control inputs, such as 2D trajectories, 3D trajectories, and camera poses, and outputs videos with a resolution of 320×576 and a length of 14 frames.

Evaluation Datasets. (1) DAVIS: To evaluate the effectiveness of ObjCtrl-2.5D on both 2D trajectories with depth and 3D trajectories, we extend the DAVIS dataset [31] by generating 3D trajectories using SpatialTracker [59]. The DAVIS dataset comprises 90 real-world videos with corresponding instance mask annotations. For each video, we use the first frame as the conditional image input for image-to-video (I2V) generation and randomly select one 3D trajectory within the instance mask as the guidance for object
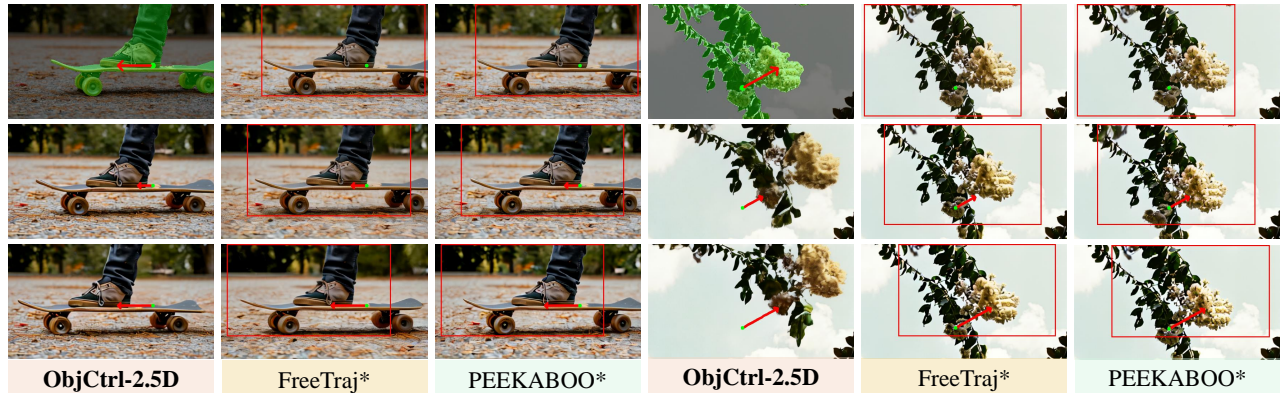
Figure 5. **Qualitative Comparison with Training-free Methods.** While PEEKABOO [19] and FreeTraj [32] can move the object coarsely within the bounding boxes generated from the trajectory, they lack control precision. In contrast, ObjCtrl-2.5D achieves higher trajectory alignment by extending the 2D trajectory to 3D and accurately transforming it into camera poses through a geometric projection algorithm (triangulation [38, 39]).
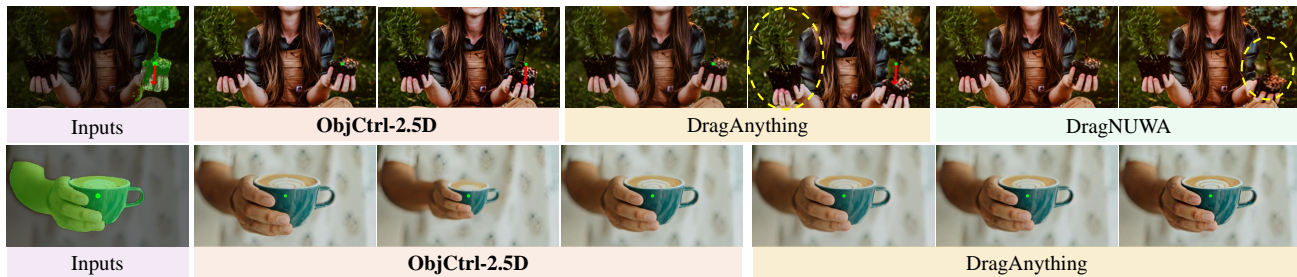

Figure 6. **Qualitative Comparison with Training-based Methods.** Due to their training strategy, DragAnything [58] tends to apply global movement to objects (both potted plants shift downward, despite only the right plant being specified to move), and DragNUWA [64] often moves only part of the target object. In contrast, our proposed ObjCtrl-2.5D achieves precise, targeted object control thanks to its Layer Control Module. Additionally, ObjCtrl-2.5D is capable of performing more versatile object control when given a trajectory with a fixed spatial position (the green point in the second sample), such as front-to-back-to-front movement, while DragAnything [58] generates a relatively static video.

control. **(2) ObjCtrl-Test:** As object movement trajectories extracted from real videos with existing trackers [20, 59] often reflect camera movement rather than precise object motion, we developed a new test set specifically for evaluating object motion control in T2V generation, termed ObjCtrl-Test. ObjCtrl-Test comprises 78 samples, each containing a high-quality image, an object mask indicating the target for movement, and a 2D trajectory. In contrast to DAVIS, where movements are commonly observed in real-world contexts, ObjCtrl-Test includes a variety of samples designed to prompt unconventional or rare object movements. **Evaluation Metrics.** Following previous works [54, 58], we evaluate the generated video quality using the Fréchet Inception Distance (FID) [40] and Fréchet Video Distance (FVD) [46], taking the real videos in DAVIS [31] as reference. To assess object motion control precision, we use ObjMC [54], which calculates the distance between target trajectories and the trajectories of generated videos, estimated using SpatialTracker [59]. Lower ObjMC scores indicate

better object control accuracy. For a more comprehensive evaluation, we additionally conduct a user study.

4.1. Comparison with State-of-the-art Methods

To provide a thorough evaluation, we compare ObjCtrl-2.5D with both training-free and training-based methods. For training-free approaches, we use two recent methods: PEEKABOO [19] and FreeTraj [32]. These methods, initially designed for I2V generation, incorporate adaptive attention mechanisms for object motion control. In adapting them for I2V generation, we omit manipulations on cross-attention since SVD [3] utilizes a single embedding feature from the conditional image for cross-attention input. We denote these adapted versions as PEEKABOO* and FreeTraj*. For training-based methods, we use Drag-NUWA [64] and DragAnything [58], both of which were trained with 2D trajectories and perform well under such conditions.

The quantitative results in Table 1 demonstrate that

Table 1. **Quantitative Comparisons on DAVIS [31] and ObjCtrl-Test**. ObjCtrl-2.5D, as a training-free approach, shows promising improvement in object motion control compared to prior training-free methods, PEEKABOO [19] and FreeTraj [32], as indicated by ObjMC scores. Although there remains room for improvement compared to training-based methods such as Drag-NUWA [64] and DragAnything [58], ObjCtrl-2.5D offers more versatile object control, as demonstrated in Figure 1 and Figure 6.

	DAVIS			ObjCtrl-Test		
Methods	FID ↓	FVD ↓	ObjMC ↓	FID ↓	FVD ↓	ObjMC ↓
DragNUWA [64]	62.36	11.68	37.57	235.94	27.45	58.80
DragAnything [58]	59.81	11.05	46.10	227.72	26.93	60.81
PEEKABOO* [19]	62.43	11.97	128.05	250.68	27.54	164.40
FreeTraj* [32]	69.72	12.62	125.30	244.88	26.74	158.39
ObjCtrl-2.5D	59.77	12.22	91.42	247.48	27.82	120.37

ObjCtrl-2.5D improves object motion control, as evidenced by the substantial reduction in the ObjMC score compared to other training-free methods. This improvement primarily stems from the fundamental differences in model design between ObjCtrl-2.5D and PEEKABOO* and FreeTraj*. Both PEEKABOO* and FreeTraj* rely on 2D trajectories represented as a series of bounding boxes, as illustrated in Figure 5. This approach enables coarse object movement within the specified bounding boxes but lacks the precision of exact trajectory alignment. In contrast, ObjCtrl-2.5D achieves higher trajectory alignment by extending the 2D trajectory to 3D and accurately transforming it into camera poses through a geometric projection algorithm (triangulation [38, 39]), yielding significantly better alignment with the given trajectory than PEEKABOO* and FreeTraj*.

On the other hand, Table 1 indicates that ObjCtrl-2.5D remains room for improvement compared to training-based methods like DragNUWA [64] and DragAnything [58]. These methods, trained on optical flow-based or tracker-derived trajectories, are inherently skilled at closely following specified trajectories, leading to high ObjMC performance. However, their design often results in moving the entire scene rather than isolating the target object's motion. This limitation is visible in DragAnything [58] in the first row of Figure 6, where both potted plants shift downward, despite only the right plant being specified to move. Moreover, in this example, DragNUWA [64] fails to move the entire right-side plant, likely due to a lack of semantic awareness. In contrast, ObjCtrl-2.5D achieves targeted object control advanced from the proposed Layer Control Module, which restricts the camera poses derived from the given trajectory to areas around the target object, minimally affecting the background. As demonstrated in the second row of Figure 6, when given a trajectory with a fixed spatial position, ObjCtrl-2.5D can perform front-to-back-to-front object movement by leveraging depth information (indicating an increase and subsequent decrease in depth). Meanwhile, DragAnything [58] tends to maintain object static in

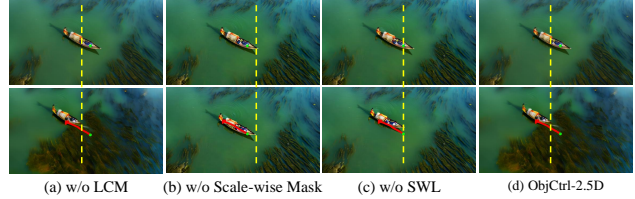| (a) w/o LCM | (b) w/o Scale-wise Mask | (c) w/o SWL | (d) ObjCtrl-2.5D |

Figure 7. **Qualitative Results of Ablation Studies on LCM, Scale-wise Mask, and SWL.** Without the Layer Control Module (LCM), ObjCtrl-2.5D applies motion control to the entire scene (a) rather than isolating the specific object (d). Removing the Shared Warping Latent (SWL) reduces controllability (c), while omitting the scale-wise mask may eliminate controllability (b).
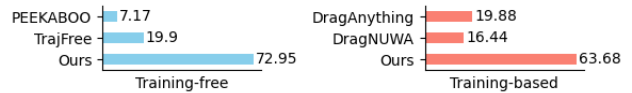

Figure 8. **User Study.** The majority of participants preferred the results obtained with ObjCtrl-2.5D over both training-free and training-based methods, attributing this preference to its better trajectory alignment and more natural motion generation.

the generated video under similar conditions.

To provide a comprehensive evaluation, we conducted a user study using the ObjCtrl-Test dataset. Fifty individuals with experience in AIGC participated, voting on which videos demonstrate better alignment of a specified object to the given trajectory and contain more natural performance. As shown in Figure 8, approximately 72.95% of participants preferred ObjCtrl-2.5D over PEEKABOO [19] and FreeTraj [32], while 63.68% favored ObjCtrl-2.5D over DragNUWA [64] and DragAnything [58] for its more natural motion generation.

4.2. Ablation Study

The effectiveness of Depth from I_c. To evaluate the effectiveness of extending a 2D trajectory to 3D using depth information from the conditional image I_c, we compare the results of ObjCtrl-2.5D's conducted on 2D trajectory with depth to results obtained using 3D trajectories in DAVIS [31], where trajectories are extracted from real-world videos. ObjCtrl-2.5D with 3D trajectories achieves an ObjMC score of 92.08, closely matching the 91.42 score obtained by combining a 2D trajectory with depth from I_c. This result indicates that supplementing a 2D trajectory with depth from I_c can effectively approximate a 3D trajectory, making it valuable for aiding object motion control in T2V generation.

The Effectiveness of Layer Control Module and Scale-wise Masks. The LCM is designed to adapt the camera motion control module for object motion control by separating the object from the background, enabling independent motion control for each. Without LCM, the base model of

Figure 9. **Qualitative Results of Ablation Studies on SWL and Copy-pasting Shared Latent.** The Shared Warping Latent (SWL) in ObjCtrl-2.5D restricts the shared latent specifically within the object's warping areas, effectively avoiding unintended effects on the background while controlling the target object. In contrast, the copy-pasting mechanism used in Free-Traj [32] coarsely applies the shared latent within bounding boxes, resulting in pronounced artifacts in the generated video.

ObjCtrl-2.5D typically aligns the trajectory by shifting the entire scene, as shown in Figure 7 (a). With LCM, however, the global motion can be segmented into two distinct camera poses for the object and background. Yet, because the features of these two camera poses are spatially merged based on object size, there is a potential risk of losing control over the object's motion. To address this, we introduce scale-wise masks that progressively dilate the merging mask as the feature scale is downsampled.

To assess the effectiveness of the scale-wise mask, we remove the dilation operation and apply the same mask at all scales. This results in an increase in ObjMC score on ObjCtrl-Test from 120.37 to 124.37 (smaller score is better). The failed object motion for the boat, as shown in Figure 7 (b), highlights this limitation. In contrast, ObjCtrl-2.5D with scale-wise masks successfully drives the target object, as seen in (c) and (d), demonstrating the effectiveness of both the LCM and scale-wise masking.

The effectiveness of Shared Warping Latent. As shown in Figure 7 (c) and (d), ObjCtrl-2.5D aligns with the given trajectories more accurately when using SWL compared to settings without it. By sharing latent across frames within the warping object areas, SWL provides strong motion guidance, enhancing trajectory accuracy. In comparison to FreeTraj's copy-and-pasting mechanism [32], where shared latent is bounded by a box that includes areas outside the object, SWL achieves a better ObjMC score (120.37 vs. 138.22) and avoids visible artifacts, as illustrated in Figure 9. *However, as with [32], we find that sharing latent across frames can decrease generation quality and is sensitive to sample variations. Given ObjCtrl-2.5D's robust object motion control with LCM, we recommend using SWL as an enhancement for more challenging cases, ensuring a balance between precise motion control and high-quality video generation.*

4.3. More Extensions

Control with Customized Camera Poses. ObjCtrl-2.5D not only accepts 2D or 3D trajectories as object motion con-

Figure 10. **Additional Results with User-Defined Camera Poses.** ObjCtrl-2.5D allows both the object and background to be manipulated using user-defined camera poses, enabling effects like zooming in, as shown in these examples. More results can be found in the supplementary materials.
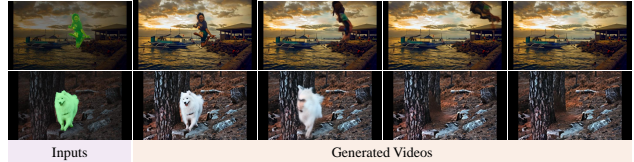

Inputs   Generated Videos

Figure 11. **Failure Cases.** Due to the limitations of SVD [3] in handling large motions, ObjCtrl-2.5D with *high-speed* camera poses results in the object fading out of the scene, leaving only the background. Interestingly, this outcome reveals potential for *image inpainting* applications, as seen in the last frames of the generated videos.

trol conditions, but also directly accepts customized camera poses, enabling even more versatile object motion control. As shown in Figure 1, given a sequence of anti-clockwise or self-rotating camera poses, ObjCtrl-2.5D can generate videos with spatial rotations (*e.g.*, the snowboarder in the second row) or 3D space rotations (*e.g.*, the rose in the third row). Additionally, more examples, such as zooming in on the object or background, are provided in Figure 10. More results can be found in the supplementary materials.

Flexible Background Movement. The LCM in ObjCtrl-2.5D enables flexible control over background motion by applying different camera poses to background areas. This includes fixed camera poses ($[\mathbf{I}|\mathbf{0}]$) across all frames, poses reversed relative to the object's movement, or no camera poses at all. Detailed visual results can be found in the supplementary materials.

4.4. Limitation

As a training-free method, the quality and motion fidelity of ObjCtrl-2.5D depends on the performance of the underlying video generation model. Since the SVD model struggles with fast-moving objects, ObjCtrl-2.5D is less effective for long trajectories within 14 frames. This limitation can lead to issues such as motion blur, misalignment, or object elimination when handling rapid or complex object movements. Figure 11 demonstrates how high-speed camera poses can cause the object to fade out of the scene, leaving only the background. Interestingly, this unintended outcome reveals potential for image inpainting applications (see the last frame).

5. Conclusion

In this study, we introduce ObjCtrl-2.5D, a novel framework designed to improve object motion control in video generation by incorporating 3D trajectories derived from 2D trajectories and scene depth information. By representing object movement through camera poses, ObjCtrl-2.5D effectively leverages existing Camera Motion Control T2V (CMC-T2V) models to achieve accurate object control without additional training. Our approach includes the development of a Layer Control Module (LCM) to isolate the target object from the background and a Shared Warping Latent (SWL) to enhance control precision by establishing consistent initial object movement. Experimental results demonstrate that ObjCtrl-2.5D largely surpasses existing training-free methods in control accuracy, as validated by both objective and subjective metrics. Additionally, ObjCtrl-2.5D supports complex object movements, such as object rotation, further broadening its application in video generation. This work underscores the value of integrating depth information for realistic video outputs and highlights the potential for future advancements in controllable 3D video generation.

References

[1] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Yuanzhen Li, Tomer Michaeli, et al. Lumiere: A space-time diffusion model for video generation. *arXiv preprint arXiv:2401.12945*, 2024. 1, 2

[2] Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. ZoeDepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023. 4

[3] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable Video Diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127*, 2023. 1, 2, 3, 5, 6, 8

[4] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023. 2

[5] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 1, 2

[6] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. VideoCrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023. 2

[7] Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. Videocrafter2:

[8] Overcoming data limitations for high-quality video diffusion models. In *CVPR*, 2024. 1, 2

[8] Tsai-Shien Chen, Chieh Hubert Lin, Hung-Yu Tseng, Tsung-Yi Lin, and Ming-Hsuan Yang. Motion-conditioned diffusion model for controllable video synthesis. *arXiv preprint arXiv:2304.14404*, 2023. 3

[9] Weifeng Chen, Jie Wu, Pan Xie, Hefeng Wu, Jiashi Li, Xin Xia, Xuefeng Xiao, and Liang Lin. Control-A-Video: Controllable text-to-video generation with diffusion models. *arXiv preprint arXiv:2305.13840*, 2023. 3

[10] Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. LucidDreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384*, 2023. 5

[11] Aidan Clark, Jeff Donahue, and Karen Simonyan. Adversarial video generation on complex datasets. *arXiv preprint arXiv:1907.06571*, 2019. 2

[12] Zuozhuo Dai, Zhenghao Zhang, Yao Yao, Bingxue Qiu, Siyu Zhu, Long Qin, and Weizhi Wang. AnimateAnything: Fine-grained open domain image animation with motion guidance. *arXiv preprint arXiv:2311.12886*, 2023. 3

[13] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. AnimateDiff: Animate your personalized text-to-image diffusion models without specific tuning. *ICLR*, 2024. 2

[14] Hao He, Yinghao Xu, Yuwei Guo, Gordon Wetzstein, Bo Dai, Hongsheng Li, and Ceyuan Yang. Cameractrl: Enabling camera control for text-to-video generation. *arXiv preprint arXiv:2404.02101*, 2024. 2, 3, 4, 5

[15] Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. Latent video diffusion models for high-fidelity video generation with arbitrary lengths. *arXiv preprint arXiv:2211.13221*, 2022. 1, 2

[16] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *NeuIPS*, 2022. 2

[17] Wenyi Hong, Ming Ding, Wendi Zheng, Xinghan Liu, and Jie Tang. CogVideo: Large-scale pretraining for text-to-video generation via transformers. *arXiv preprint arXiv:2205.15868*, 2022. 1, 2

[18] Chen Hou, Guoqiang Wei, Yan Zeng, and Zhibo Chen. Training-free camera control for video generation. *arXiv preprint arXiv:2406.10126*, 2024. 5

[19] Yash Jain, Anshul Nasery, Vibhav Vineet, and Harkirat Behl. PEEKABOO: Interactive video generation via masked-diffusion. In *CVPR*, 2024. 1, 2, 3, 4, 6, 7, 12, 14

[20] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In *ECCV*, 2024. 6

[21] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023. 5

[22] Yaowei Li, Xintao Wang, Zhaoyang Zhang, Zhouxia Wang, Ziyang Yuan, Liangbin Xie, Yuexian Zou, and Ying Shan. Image Conductor: Precision control for interactive video synthesis. *arXiv preprint arXiv:2406.15339*, 2024. 1, 3

[23] Wan-Duo Kurt Ma, JP Lewis, and W Bastiaan Kleijn. Trailblazer: Trajectory control for diffusion-based video generation. *arXiv preprint arXiv:2401.00896*, 2023. 1, 2, 3

[24] Xin Ma, Yaohui Wang, Gengyun Jia, Xinyuan Chen, Ziwei Liu, Yuan-Fang Li, Cunjian Chen, and Yu Qiao. Latte: Latent diffusion transformer for video generation. *arXiv preprint arXiv:2401.03048*, 2024. 1, 2

[25] Chong Mou, Mingdeng Cao, Xintao Wang, Zhaoyang Zhang, Ying Shan, and Jian Zhang. ReVideo: Remake a video with motion and content control. *NeuIPS*, 2024. 3

[26] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. DragonDiffusion: Enabling drag-style manipulation on diffusion models. *ICLR*, 2024. 3

[27] Koichi Namekata, Sherwin Bahmani, Ziyi Wu, Yash Kant, Igor Gilitschenski, and David B Lindell. SG-I2V: Self-guided trajectory control in image-to-video generation. *arXiv preprint arXiv:2411.04989*, 2024. 3

[28] Katsunori Ohnishi, Shohei Yamamoto, Yoshitaka Ushiku, and Tatsuya Harada. Hierarchical video generation from orthogonal information: Optical flow and texture. In *AAAI*, 2018. 2

[29] Hao Ouyang, Kathryn Heal, Stephen Lombardi, and Tiancheng Sun. Text2Immersion: Generative immersive scene with 3d gaussians. *arXiv preprint arXiv:2312.09242*, 2023. 5

[30] Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: Universal monocular metric depth estimation. In *CVPR*, 2024. 4

[31] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 2, 5, 6, 7

[32] Haonan Qiu, Zhaoxi Chen, Zhouxia Wang, Yingqing He, Menghan Xia, and Ziwei Liu. FreeTraj: Tuning-free trajectory control in video diffusion models. *arXiv preprint arXiv:2406.16863*, 2024. 1, 2, 3, 4, 5, 6, 7, 8, 12, 14

[33] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 5

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2, 3

[36] Masaki Saito and Shunta Saito. TGANv2: Efficient training of large models for video generation with multiple subsampling layers. *arXiv preprint arXiv:1811.09245*, 2018. 2

[37] Masaki Saito, Eiichi Matsumoto, and Shunta Saito. Temporal generative adversarial nets with singular value clipping. In *ICCV*, 2017. 2

[38] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-Motion revisited. In *CVPR*, 2016. 2, 4, 6, 7

[39] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 2, 4, 6, 7

[40] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, 2020. 6

[41] Junyoung Seo, Kazumi Fukuda, Takashi Shibuya, Takuya Narihira, Naoki Murata, Shoukang Hu, Chieh-Hsin Lai, Seungryong Kim, and Yuki Mitsufuji. GenWarp: Single image to novel views with semantic-preserving generative warping. *arXiv preprint arXiv:2405.17251*, 2024. 5

[42] Yujun Shi, Chuhui Xue, Jun Hao Liew, Jiachun Pan, Hanshu Yan, Wenqing Zhang, Vincent YF Tan, and Song Bai. DragDiffusion: Harnessing diffusion models for interactive point-based image editing. In *CVPR*, 2024. 3

[43] Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light Field Networks: Neural scene representations with single-evaluation rendering. *NeuIPS*, 2021. 3

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

[45] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *CVPR*, 2018. 2

[46] Thomas Unterthiner, Sjoerd Van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 6

[47] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. *NeuIPS*, 2016. 2

[48] Jiuniu Wang, Hangjie Yuan, Dayou Chen, Yingya Zhang, Xiang Wang, and Shiwei Zhang. ModelScope text-to-video technical report. *arXiv preprint arXiv:2308.06571*, 2023. 1, 2

[49] Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. Boximator: Generating rich and controllable motions for video synthesis. *arXiv preprint arXiv:2402.01566*, 2024. 1, 3

[50] Qianqian Wang, Vickie Ye, Hang Gao, Jake Austin, Zhengqi Li, and Angjoo Kanazawa. Shape of Motion: 4d reconstruction from a single video. 2024. 4

[51] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3d vision made easy. In *CVPR*, 2024. 4

[52] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. VideoComposer: Compositional video synthesis with motion controllability. In *NeuIPS*, 2023. 3

[53] Yaohui Wang, Piotr Bilinski, Francois Bremond, and Antitza Dantcheva. G3an: Disentangling appearance and motion for video generation. In *CVPR*, 2020. 2

[54] 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 *SIGGRAPH*, 2024. 1, 2, 3, 6

[55] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Stan Weixian Lei, Yuchao Gu, Yufei Shi, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-A-Video: One-shot tuning of image diffusion models for text-to-video generation. In *ICCV*, 2023. 3

[56] Ruiqi Wu, Liangyu Chen, Tong Yang, Chunle Guo, Chongyi Li, and Xiangyu Zhang. LAMP: Learn a motion pattern for few-shot-based video generation. *arXiv preprint arXiv:2310.10769*, 2023. 3

[57] Tianxing Wu, Chenyang Si, Yuming Jiang, Ziqi Huang, and Ziwei Liu. FreeInit: Bridging initialization gap in video diffusion models. *arXiv preprint arXiv:2312.07537*, 2023. 5

[58] Weijia Wu, Zhuang Li, Yuchao Gu, Rui Zhao, Yefei He, Junhao David Zhang, Shou Mike Zheng, Yan Li, Tingting Gao, and Di Zhang. DragAnything: Motion control for anything using entity representation. In *ECCV*, 2024. 1, 2, 3, 4, 6, 7, 12, 14

[59] Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. Spatialtracker: Tracking any 2d pixels in 3d space. In *CVPR*, 2024. 4, 5, 6

[60] Zeqi Xiao, Yifan Zhou, Shuai Yang, and Xingang Pan. Video diffusion models are training-free motion interpreter and controller. *NeuIPS*, 2024. 3

[61] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. DynamiCrafter: Animating open-domain images with video diffusion priors. In *ECCV*, 2025. 1, 2

[62] Shiyuan Yang, Liang Hou, Haibin Huang, Chongyang Ma, Pengfei Wan, Di Zhang, Xiaodong Chen, and Jing Liao. Direct-a-Video: Customized video generation with user-directed camera movement and object motion. In *SIGGRAPH*, 2024. 1, 3

[63] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 1, 2

[64] Shengming Yin, Chenfei Wu, Jian Liang, Jie Shi, Houqiang Li, Gong Ming, and Nan Duan. DragNUWA: Fine-grained control in video generation by integrating text, image, and trajectory. *arXiv preprint arXiv:2308.08089*, 2023. 1, 3, 4, 6, 7, 12, 14

[65] Yan Zeng, Guoqiang Wei, Jiani Zheng, Jiaxin Zou, Yang Wei, Yuchen Zhang, and Hang Li. Make Pixels Dance: High-dynamic video generation. In *CVPR*, 2024. 1, 2

[66] Rui Zhao, Yuchao Gu, Jay Zhangjie Wu, David Junhao Zhang, Jiawei Liu, Weijia Wu, Jussi Keppo, and Mike Zheng Shou. MotionDirector: Motion customization of text-to-video diffusion models. *ECCV*, 2024. 3

[67] Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. Open-Sora: Democratizing efficient video production for all, march 2024. *URL https://github.com/hpcaitech/Open-Sora*. 1, 2

ObjCtrl-2.5D: Training-free Object Control with Camera Poses

Project page: https://wzhouxiff.github.io/projects/ObjCtrl-2.5D/

Supplementary Material

The supplementary materials provide additional details and results achieved with the proposed ObjCtrl-2.5D, accompanied by in-depth analyses. **For a comprehensive understanding, we highly encourage readers to view the project page showcasing dynamic results.** The structure of the supplementary materials is outlined as follows:

- Section A provides additional details on transforming 2D trajectories into 3D using depth extracted from the conditional image.
- Section B discusses extensions involving customized camera poses and flexible background movements.
- Section C presents additional comparative results with previous methods.
- Section D showcases more results generated using ObjCtrl-2.5D.

A. More Details about 2D Trajectories to 3D

In this work, ObjCtrl-2.5D extends 2D trajectories to 3D by utilizing depth information, $\mathbf{D_c}$, extracted from the conditional image $\mathbf{I_c}$. The depth d^i of each trajectory point (x^i, y^i) is determined by the corresponding depth value $\mathbf{D_c}(x^i, y^i)$. When the trajectory spans both the foreground object and the background, significant depth variations may occur between consecutive points, as shown in Figure 12 (a). This can result in abrupt changes in object movement along the trajectory. To address this, we smooth the 3D trajectory by analyzing its gradient, defined as $grad = d^i - d^{i-1}, i \in [1, N-1]$, and computing the standard deviation of the gradient, $grad_{std} = \mathbf{std}(grad)$. If $grad_{std} > \theta$, the depth d^i is reset to the initial depth d^0. In this work, we set $\theta = 0.2$.

To prevent such issues, we recommend drawing the trajectory directly on the depth image, as shown in Figure 12, which inherently provides smoother depth transitions ((b) and (c)) and avoids the abrupt changes shown in (a). Additionally, unlike previous methods such as Drag-NUWA [64] and DragAnything [58], which require trajectories to start specifically from the target object, ObjCtrl-2.5D offers greater flexibility. Trajectories can be drawn anywhere on the depth image, ensuring a suitable depth value for each point. **This flexibility is achieved because the trajectory in ObjCtrl-2.5D serves only to indicate object motion and is ultimately transformed into spatially independent camera poses.** Object-specific motion is then implemented using the merged mask introduced by the Layer Control Module in ObjCtrl-2.5D.

B. More Extensions

Object Control with Customized Camera Poses. ObjCtrl-2.5D supports user-defined camera poses for controlling the motion of objects or the background. Beyond the "Zoom In" camera poses presented in the main manuscript, we showcase additional results using various camera poses, including zoom out, pan left, and pan right, as illustrated in Figure 13. The examples demonstrate that ObjCtrl-2.5D can drive the same sample differently with different camera poses, such as the leftward, rightward, and forward movements of the cloud in the second example.

C. More Compared Results

We provide additional comparisons with previous methods. As shown in Figure 14, ObjCtrl-2.5D outperforms the training-free methods, including PEEKABOO [19] and FreeTraj [32], in trajectory alignment. While training-based methods like DragNUWA [64] and DragAnything [58] also achieve good trajectory alignment, they often rely on global movement or parts of the object movement rather than targeting the specific object. In contrast, ObjCtrl-2.5D incorporates a Layer Control Module, enabling relatively precise control over the specific object with minimal impact on other areas of the scene, while maintaining natural video generation. **We strongly recommend viewing the project page for dynamic results.**

D. More Results of ObjCtrl-2.5D

In Figure 15, we present additional results highlighting the versatility of ObjCtrl-2.5D in object motion control, achieved through a wide range of trajectories and camera poses. *Notably, with the same rotating camera poses, ObjCtrl-2.5D can produce either self-rotation of the object (first column of Figure 15 (b)) or 3D spatial rotation (second and third columns of Figure 15 (b)), depending on the object's spatial location within the input image.* Additionally, ObjCtrl-2.5D allows fine adjustments to object motion speed by altering camera movements. For instance, as shown in the fourth and fifth columns of Figure 15 (b), the car in the input image moves left at varying speeds based on different Pan Left camera poses.
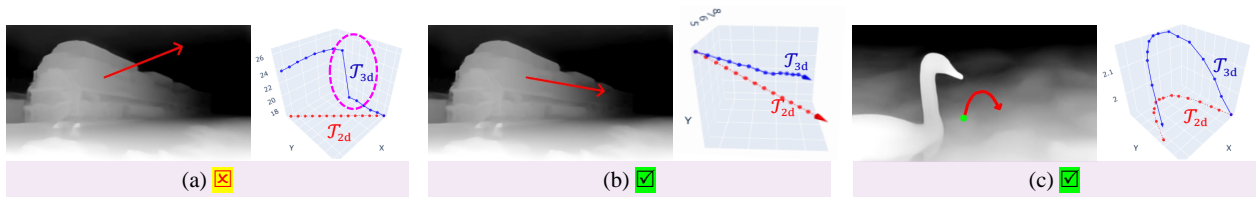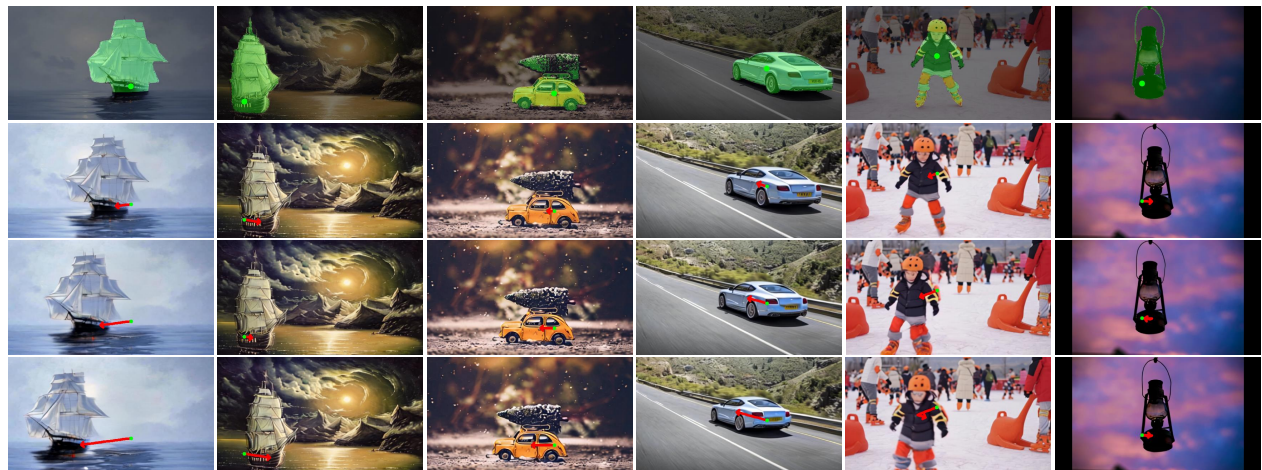
Figure 12. **Guidelines for Drawing Trajectories.** Drawing 2D trajectories directly on the depth image is recommended, as it ensures smoother depth transitions and avoids abrupt changes (refer to (a)) with the intrinsic depth information. Furthermore, trajectories can be drawn anywhere on the depth image to achieve appropriate depth values without affecting the movement of the target object.
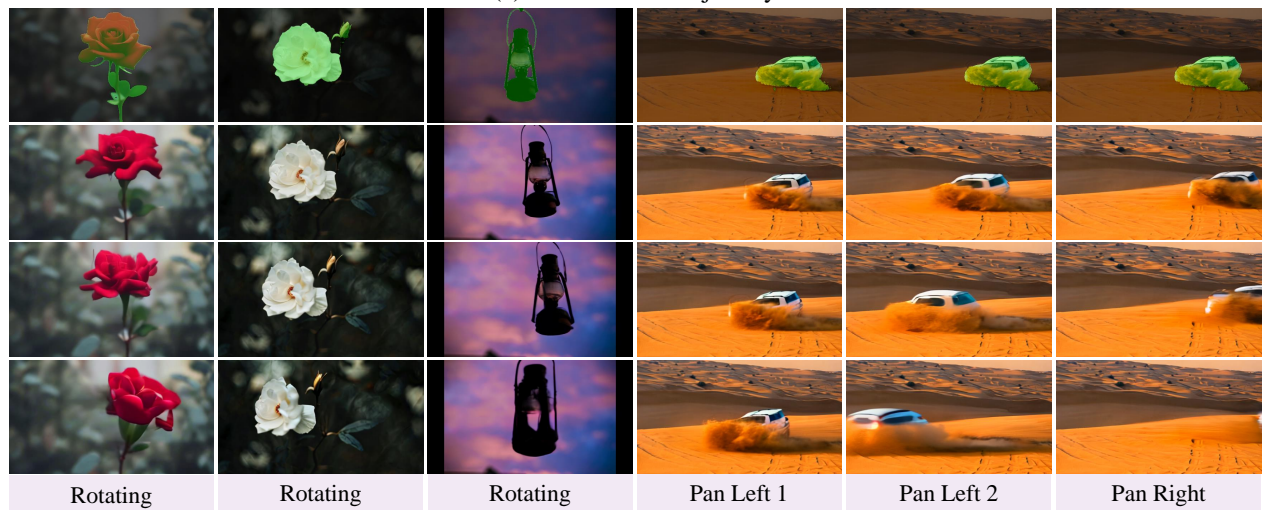

Figure 13. **Additional Results with User-Defined Camera Poses.** ObjCtrl-2.5D can drive the same sample differently with different camera poses. **We strongly recommend viewing the project page for dynamic results.**

13

| Inputs | **ObjCtrl-2.5D** | PEEKABOO* | FreeTraj* | DragNUWA | DragAnything |

Figure 14. **More Compared Results with Previous Methods.** ObjCtrl-2.5D outperforms training-free methods (PEEKABOO [19] and FreeTraj [32]) in trajectory alignment and achieves more precise target object movement compared to training-based methods (Drag-NUWA [64] and DragAnything [58]), which often result in either global scene movement or partial object movement. **We strongly recommend viewing the project page for dynamic results.**

(a) Guided with Trajectory

| Rotating | Rotating | Rotating | Pan Left 1 | Pan Left 2 | Pan Right |

(b) Guided with Camera Poses Directly

Figure 15. **More Results of ObjCtrl-2.5D.** ObjCtrl-2.5D supports a wide range of trajectories and camera poses, showcasing its versatility in object motion control. **We strongly recommend viewing the project page for dynamic results.**