Diffusion as Shader: 3D-aware Video Diffusion for Versatile Video Generation Control

ZEKAI GU, Hong Kong University of Science and Technology, China
RUI YAN, Zhejiang University, China
JIAHAO LU, Hong Kong University of Science and Technology, China
PENG LI, Hong Kong University of Science and Technology, China
ZHIYANG DOU, The University of Hong Kong, China
CHENYANG SI, Nanyang Technological University, Singapore
ZHEN DONG, Wuhan University, China
QIFENG LIU, Hong Kong University of Science and Technology, China
CHENG LIN, The University of Hong Kong, China
ZIWEI LIU, Nanyang Technological University, Singapore
WENPING WANG, Texas A&M University, U.S.A
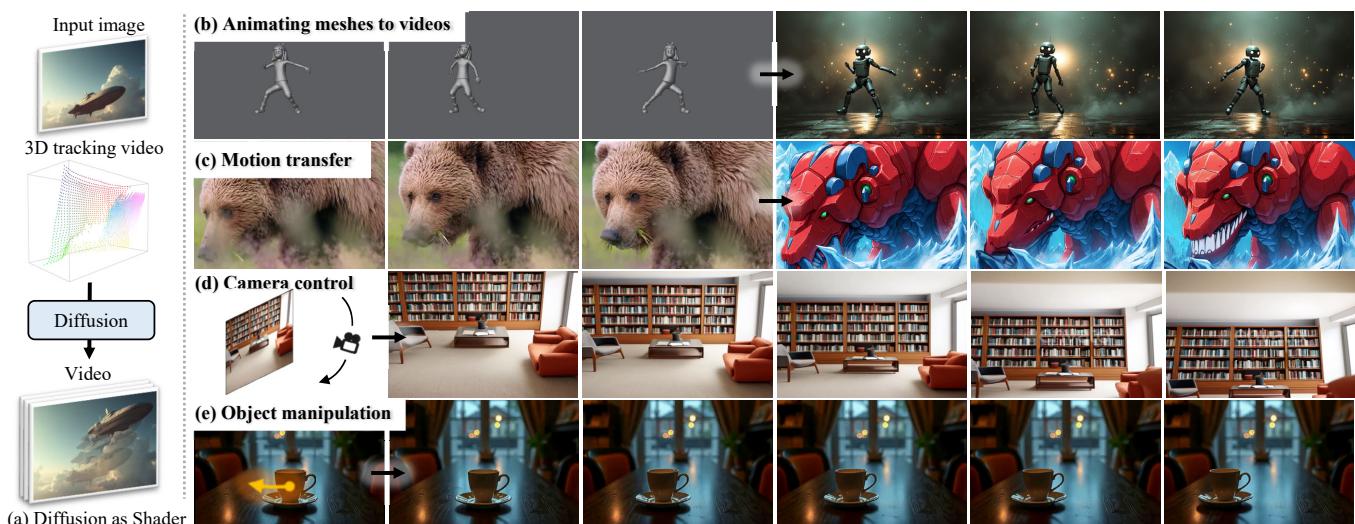YUAN LIU, Hong Kong University of Science and Technology, China

Fig. 1. **Diffusion as Shader (DaS)** is (a) a 3D-aware video diffusion method enabling versatile video control tasks including (b) animating meshes to video generation, (c) motion transfer, (d) camera control, and (e) object manipulation.

Diffusion models have demonstrated impressive performance in generating high-quality videos from text prompts or images. However, precise control over the video generation process—such as camera manipulation or content editing—remains a significant challenge. Existing methods for controlled video generation are typically limited to a single control type, lacking the flexibility to handle diverse control demands. In this paper, we introduce Diffusion as Shader (DaS), a novel approach that supports multiple video control tasks within a unified architecture. Our key insight is that achieving versatile video control necessitates leveraging 3D control signals, as videos are fundamentally 2D renderings of dynamic 3D content. Unlike prior methods limited to 2D control signals, DaS leverages 3D tracking videos as control inputs, making the video diffusion process inherently 3D-aware. This innovation allows DaS to achieve a wide range of video controls by simply manipulating the 3D tracking videos. A further advantage of using 3D tracking videos is their ability to effectively link frames, significantly enhancing the temporal consistency of the generated videos. With just 3 days of fine-tuning on 8 H800 GPUs using less than 10k videos, DaS demonstrates strong control capabilities across diverse tasks, including mesh-to-video generation, camera control, motion transfer, and object manipulation. Codes and more results are available at https://igl-hkust.github.io/das/.

1 INTRODUCTION

The development of diffusion generative models [Blattmann et al. 2023; Brooks et al. 2024; Ho et al. 2020; Lin et al. 2024; Rombach et al. 2022; Zheng et al. 2024b] enables high-quality video generation from text prompts or a starting image. Recent emerging models, e.g. Sora [Brooks et al. 2024], CogVideo-X [Yang et al. 2024b], Kel-ing [Kuaishou 2024], and Hunyuan [Kong et al. 2024], have shown impressive video generation ability with strong temporal consistency and appealing visual effects, which becomes a promising tool for artists to create stunning videos using just few images or text

prompts. These advancements show strong potential to revolutionize the advertising, film, robotics, and game industries, becoming fundamental elements for various generative AI-based applications.

A major challenge in video generation lies in achieving versatile and precise control to align seamlessly with users' creative visions. While recent methods have introduced strategies to integrate control into the video generation process [Guo et al. 2024; He et al. 2024b,a; Huang et al. 2023; Ma et al. 2024b,a; Namekata et al. 2024; Polyak et al. 2024; Wang et al. 2024f,c; Yuan et al. 2024], they predominantly focus on specific control types, relying on specialized architectures that lack adaptability to emerging control requirements. Furthermore, these approaches are generally limited to high-level adjustments—such as camera movements or maintaining identity—falling short when it comes to enabling fine-grained modifications, like precisely raising an avatar's left hand.

We argue that achieving versatile and precise video generation control fundamentally requires 3D control signals in the diffusion model. Videos are 2D renderings of dynamic 3D content. In a traditional Computer Graphics (CG)- based video-making pipeline, we can effectively control all aspects of a video in detail by manipulating the underlying 3D representations, such as meshes or particles. However, existing video control methods solely apply 2D control signals on rendered pixels, lacking the 3D awareness in the video generation process and thus struggling to achieve versatile and fine-grained controls. Thus, to this end, we present a novel 3D-aware video diffusion method, called **Diffusion as Shader** (DaS) in this paper, which utilizes 3D control signals to enable diverse and precise control tasks within a unified architecture.

Specifically, as shown in Figure 1 (a), DaS is an image-to-video diffusion model that takes a 3D tracking video as the 3D control signals for various control tasks. The 3D tracking video contains the motion trajectories of 3D points whose colors are defined by their coordinates in the camera coordinate system of the first frame. In this way, the 3D tracking video represents the underlying 3D motion of this video. The video diffusion model acts like a shader to compute shaded appearances on the dynamic 3D points to generate the video. Thus, we call our model *Diffusion as Shader*.

Using 3D tracking videos as control signals offers a significant advantage over depth videos with enhanced temporal consistency. While a straightforward approach to incorporating 3D control into video diffusion models involves using depth maps as control signals, depth maps only define the structural properties of the underlying 3D content without explicitly linking frames across time. In contrast, 3D tracking videos provide a consistent association between frames, as identical 3D points maintain the same colors across the video. These color anchors ensure consistent appearances for the same 3D points, thereby significantly improving temporal coherence in the generated videos. Our experiments demonstrate that even when a 3D region temporarily disappears and later reappears, DaS effectively preserves the appearance consistency of that region, thanks to the temporal consistency enabled by the tracking video.

By leveraging 3D tracking videos, DaS enables versatile video generation controls, encompassing but not limited to the following video control tasks.

(1) *Animating meshes to videos.* Using advanced 3D tools like Blender, we can design animated 3D meshes based on predefined templates. These animated meshes are transformed into 3D tracking videos to guide high-quality video generation (Figure 1 (b)).

(2) *Motion transfer.* Starting with an input video, we employ a 3D tracker [Xiao et al. 2024b] to generate a corresponding 3D tracking video. Next, the depth-to-image Flux model [Labs 2024] is used to modify the style or content of the first frame. Based on the updated first frame and the 3D tracking video, DaS generates a new video that replicates the motion patterns of the original while reflecting the new style or content (Figure 1 (c)).

(3) *Camera control.* To enable precise camera control, depth maps are estimated to extract 3D points [Bochkovskii et al. 2024]. These 3D points are then projected onto a specified camera path to create a 3D tracking video, which guides the generation of videos with customized camera movements (Figure 1 (d)).

(4) *Object manipulation.* By integrating object segmentation techniques [Kirillov et al. 2023] with a monocular depth estimator [Bochkovskii et al. 2024], the 3D points of specific objects can be extracted and manipulated. These modified 3D points are used to construct a 3D tracking video, which guides the creation of videos for object manipulation (Figure 1 (e)).

Due to the 3D awareness of DaS, DaS is data-efficient. Finetuning with less than 10k videos on 8 H800 GPUs for 3 days already gives the powerful control ability to DaS, which is demonstrated by various control tasks. We compare DaS with baseline methods on camera control [He et al. 2024b; Wang et al. 2024c] and motion transfer [Geyer et al. 2023a], which demonstrates that DaS achieves significantly improved performances in these two controlling tasks than baselines. For the remaining two tasks, i.e. mesh-to-video and object manipulation, we provide extensive qualitative results to show the superior generation quality of our method.

2 RELATED WORK

2.1 Video diffusion

In recent years, the success of diffusion models in image generation [Ho et al. 2020; Peebles and Xie 2023a; Rombach et al. 2022] has sparked interest in exploring video generation [Blattmann et al. 2023; Brooks et al. 2024; Chen et al. 2023b, 2024b; Guo et al. 2023; He et al. 2022; Ho et al. 2022; Kong et al. 2024; Kuaishou 2024; Lin et al. 2024; Xing et al. 2024; Yang et al. 2024b; Zheng et al. 2024b]. VDM [Ho et al. 2022] is the first work to explore the feasibility of diffusion in the field of video generation. SVD [Blattmann et al. 2023] introduces a unified strategy for training a robust video generation model. Sora [Brooks et al. 2024], through training on extensive video data, suggests that scaling video generation models is a promising path towards building general-purpose simulators of the physical world. CogVideo-X [Yang et al. 2024b], VideoCrafter [Chen et al. 2023b, 2024b], DynamiCrafter [Xing et al. 2024], Keling [Kuaishou 2024], and Hunyuan [Kong et al. 2024] have demonstrated impressive video generation performance with strong temporal consistency.

Controllable video generation. Existing works still lack an effective way to control the generation process. There are many works [Guo et al. 2024; He et al. 2024b,a; Huang et al. 2023; Ma et al. 2024b,a,a; Namekata et al. 2024; Polyak et al. 2024; Qiu et al.

2024; Wang et al. 2024f,c; Yu et al. 2024; Yuan et al. 2024] that introduce a specific control signal in the video generation process which can only achieve one control type like identity preserving, camera control, and motion transfer. Our method is more versatile in various video control types by using a 3D-aware video generation with 3D tracking videos as conditions.

2.2 Controlled video generation

We review the following 4 types of controlled video generation.

Animating meshes to videos. Animating meshes to videos aims to texture meshes. Several works [Cai et al. 2024; Cao et al. 2023; Richardson et al. 2023; Wang et al. 2023] have demonstrated the feasibility of mesh texturization using powerful diffusion models. TexFusion [Cao et al. 2023] applies the diffusion model's denoiser on a set of 2D renders of the 3D object, optimizing an intermediate neural color field to output final RGB textures. TEXTure [Richardson et al. 2023] introduces a dynamic trimap representation and a novel diffusion sampling process, leveraging this trimap to generate seamless textures from various views. G-Rendering [Cai et al. 2024] takes a dynamic mesh as input. To preserve consistency, G-Rendering employs UV-guided noise initialization and correspondence-aware blending of both pre- and post-attention features. Following G-Rendering, our method also targets dynamic meshes, utilizing a diffusion model as a shader to incorporate realistic texture information. Unlike G-Rendering, which preserves consistency at the noise and attention levels, our approach leverages 3D tracking videos as supplementary information, integrating them into the diffusion model to ensure both temporal and spatial consistency.

Camera control. Camera control [Bahmani et al. 2024; Geng et al. 2024; He et al. 2024b; Wang et al. 2024e,c; Xiao et al. 2024a; Yang et al. 2024a; Yu et al. 2024; Zheng et al. 2024a] is an important capability for enhancing the realism of generated videos and increasing user engagement by allowing customized viewpoints. Recently, many efforts have been made to introduce camera control in video generation. MotionCtrl [Wang et al. 2024c] incorporates a flexible motion controller for video generation, which can independently or jointly control camera motion and object motion in generated videos. CameraCtrl [He et al. 2024b] adopts Plücker embeddings [Sitzmann et al. 2021] as the primary form of camera parameters, enabling the ViewCrafter [Yu et al. 2024] employs a point-based representation for free-view rendering, enabling precise camera control. AC3D [Bahmani et al. 2024] optimizes pose conditioning schedules during training and testing to accelerate convergence and restricts the injection of camera conditioning to specific positions, reducing interference with other meaningful video features. CPA [Wang et al. 2024e] incorporates a Sparse Motion Encoding Module to embed the camera pose information and integrating the embedded motion information via temporal attention. Our method aims to use 3D tracking videos as an intermediary to achieve precise and consistent camera control.

Motion transfer. Motion transfer [Esser et al. 2023; Geng et al. 2024; Geyer et al. 2023a; Meral et al. 2024; Park et al. 2024; Pondaven et al. 2024; Wang et al. 2024d,c; Yatim et al. 2024] aims to synthesize novel videos by following the motion of the original one. Gen-1 [Esser et al. 2023] employs depth estimation results [Bochkovskii

et al. 2024; Lu et al. 2024; Ranftl et al. 2020] to guide the motion. TokenFlow [Geyer et al. 2023a] achieves consistent motion transfer by enforcing consistency in the diffusion feature space. MotionCtrl [Wang et al. 2024c] also achieves motion transfer by incorporating a motion controller. DiTFlow [Pondaven et al. 2024] proposes Attention Motion Flow as guidance for motion transfer on DiTs [Peebles and Xie 2023a]. Motion Prompting [Geng et al. 2024] utilizes 2D motions as prompts to realize impressive motion transfer. Unlike these approaches, our method employs 3D tracking as guidance for motion transfer, enabling a more comprehensive capture of each object's motion and the relationships between them within the video. This ensures accurate and globally consistent geometric and temporal consistency.

Object manipulation. Object manipulation refers to versatile object movement control for image-to-video generation. Different from camera control, which focuses on changes in perspective, object manipulation emphasizes the movement of the objects themselves. Currently, mainstream methods [Chen et al. 2023a; Geng et al. 2024; Jain et al. 2024; Li et al. 2024; Ma et al. 2024b; Mou et al. 2024; Qiu et al. 2024; Teng et al. 2023; Wang et al. 2024f,c; Yang et al. 2024a; Yin et al. 2023] typically achieve object manipulation by utilizing directed trajectories or modeling the relationships between bounding boxes with specific semantic meanings. However, these methods primarily rely on 2D guidance to represent the spatial movement of target objects, which often fails to accurately capture user intent and frequently results in distorted outputs. ObjCtrl-2.5D [Wang et al. 2024a] tries to address this limitation by extending 2D trajectories with depth information, creating a single 3D trajectory as the control signal. Better than the single 3D trajectory, our method leverages 3D tracking videos, which offer greater details and more effectively represent the motion relationships between foreground and background for more precise and realistic object manipulation.

Concurrent works. Recently, several works [Feng et al. 2024a; Geng et al. 2024; Jeong et al. 2024; Koroglu et al. 2024; Lei et al. 2024; Niu et al. 2024; Shi et al. 2024; Zhang et al. 2024] have explored utilizing motion as control signals. These approaches can be broadly categorized into two groups: 2D motion-based and 3D motion-based methods. [Koroglu et al. 2024; Lei et al. 2024; Shi et al. 2024] leverage 2D optical flow to condition motion, while [Geng et al. 2024; Jeong et al. 2024; Niu et al. 2024] utilize 2D tracks, which are sparser than optical flow, to track or control video motion. [Zhang et al. 2024] learns to generate 3D coordinates in the video diffusion model, which 3D awareness. [Feng et al. 2024a] lifts videos into 3D space and extracts the motion of 3D points, enabling a more accurate capture of spatial relationships between objects and supporting tasks such as object manipulation and camera control. Our method, DaS, also leverages recent tracking methods [Xiao et al. 2024b; Zhang et al. 2025] to construct videos. However, we extend the applicability by unifying a broader range of control tasks, including mesh-to-video generation and motion transfer.

3 METHOD

3.1 Overview

DaS is an image-to-video (I2V) diffusion generative model, which applies both an input image and a 3D tracking video as conditions
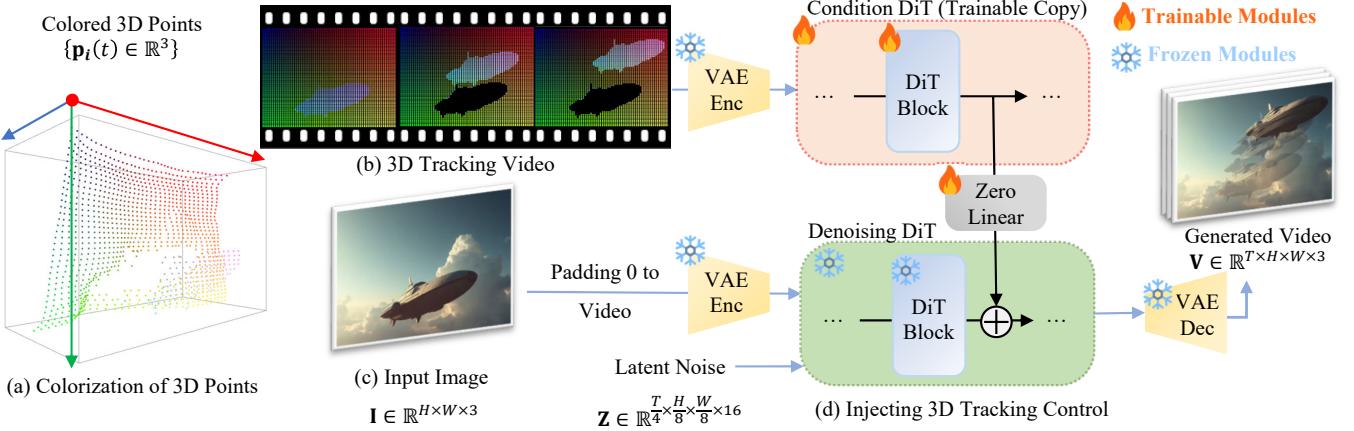
Fig. 2. **Architecture of DaS**. (a) We colorize dynamic 3D points according to their coordinates to get (b) a 3D tracking video. (c) The input image and the 3D tracking video are processed by (d) a transformer-based latent diffusion with a variational autoencoder (VAE). The 3D tracking video is processed by a trainable copy of the denoising DiT and zero linear layers are used to inject the condition features from 3D tracking videos into the denoising process.

for controllable video generation. In the following, we first review the backend I2V video diffusion model in Sec. 3.2. Then, we discuss the definition of the 3D tracking video and how to inject the 3D tracking video into the generation process as a condition in Sec. 3.3. Finally, in Sec. 3.4, we discuss how to apply DaS in various types of video generation control.

3.2 Backend video diffusion model

DaS is finetuned from the CogVideoX [Yang et al. 2024b] model that is a transformer-based video diffusion model [Peebles and Xie 2023a] operating on a latent space. Specifically, as shown in Figure 2 (d), we adopt the I2V CogVideoX model as the base model, which takes an image $I \in \mathbb{R}^{H \times W \times 3}$ as input and generate a video $V \in \mathbb{R}^{T \times H \times W \times 3}$. The generated video V has T frames with the same image size of width W height H as the input image. The input image I is first padded with zeros to get an input condition video with the same size $T \times H \times W \times 3$ as the target video. Then, a VAE encoder is applied to the padded condition video to get a latent vector of size $\frac{T}{4} \times \frac{H}{8} \times \frac{W}{8} \times 16$, which is concatenated with a noise of the same size. A diffusion transformer (DiT) [Peebles and Xie 2023b] is iteratively used to denoise the noise latent for a predefined number of steps and the output denoised latent is processed by a VAE decoder to get the video V. In the following, we discuss how to add a 3D tracking video as an additional condition on this base model.

3.3 Finetuning with 3D tracking videos

We add a 3D tracking video as an additional condition to our video diffusion model. As shown in Figure 2 (a, b), the 3D tracking video is rendered from a set of moving 3D points $\{\mathbf{p}_i(t) \in \mathbb{R}^3\}$, where $t = 1, ..., T$ means the frame index in the video. The colors of these points are determined by their coordinates in the first frame, where we normalize the coordinates into $[0, 1]^3$ and convert the coordinates into RGB colors $\{\mathbf{c}_i\}$. Note we adopt the reciprocal of z-coordinate in the normalization. These colors remain the same for different timesteps t. Then, to get a specific t-th frame of the tracking video,

we project these 3D points onto the t-th camera to render this frame. In Sec. 3.4, we will discuss how to get these moving 3D points and the camera poses of different frames for different control tasks. Next, we first introduce the architecture to utilize the 3D tracking video as a condition for video generation.

Injecting 3D tracking control. We follow a similar design as the ControlNet [Chen et al. 2024a; Zhang et al. 2023] in DaS to add the 3D tracking video as the additional condition. As shown in Figure 2 (d), we apply the pretrained VAE encoder to encode the 3D tracking video to get the latent vector. Then, we make a trainable copy of the pretrained denoising DiT, called condition DiT, to process the latent vector of the 3D tracking video. The denoising DiT contains 42 blocks and we copy the first 18 blocks as the condition DiT. In the condition DiT, we extract the output feature of each DiT block, process it with a zero-initialized linear layer, and add the feature to the corresponding feature map of the denoising DiT. We finetune the condition DiT with the diffusion losses while freezing the pretrained denoising DiT.

Finetuning details. To train the DaS model, we construct a training dataset containing both real-world videos and synthetic rendered videos. The real-world videos are from MiraData [Ju et al. 2024] while we use the meshes and motion sequences from Mixamo to render synthetic videos. All videos are center-cropped and resized to 720×480 resolution with 49 frames. We only finetune the copied condition DiT while freezing all the original denoising DiT. To construct the 3D tracking video for the rendered videos, since we have access to the ground-truth 3D meshes and camera poses for the synthetic videos, we construct our 3D tracking videos directly using these dense ground-truth 3D points, which results in dense 3D point tracking. For real-world videos, we adopt SpatialTracker [Xiao et al. 2024b] to detect 3D points and their trajectories in the 3D space. Specifically, for each real-world video, we detect 4,900 3D evenly distributed points and track their trajectories. For training, we employ a learning rate of 1×10^{-4} using the AdamW optimizer. We train the model for 2000 steps using the gradient accumulation
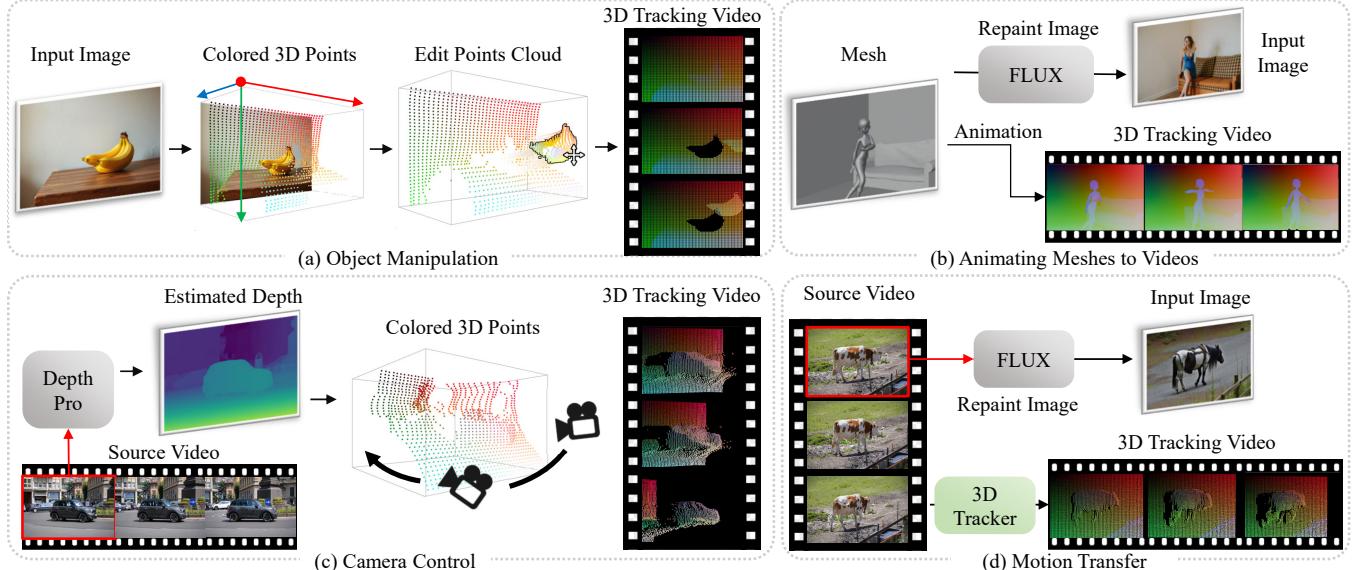
Fig. 3. **3D tracking video generation** in (a) object manipulation, (b) animating mesh to video generation, (c) camera control, and (d) motion transfer.

strategy to get an effective batch size of 64. The training takes 3 days on 8 H800 GPUs.

3.4 Video generation control

In this section, we describe how to utilize DaS for the following controllable video generation.

3.4.1 Object manipulation. DaS can generate a video to manipulate a specific object. As shown in Figure 3 (a), given an image, we estimate the depth map using Depth Pro [Bochkovskii et al. 2024] or MoGE [Wang et al. 2024b] and segment out the object using SAM [Kirillov et al. 2023]. Then, we are able to manipulate the point cloud of the object to construct a 3D tracking video for object manipulation video generation.

3.4.2 Animating meshes to videos. DaS enables the creation of visually appealing, high-quality videos from simple animated meshes. While many Computer Graphics (CG) software tools provide basic 3D models and motion templates to generate animated meshes, these outputs are often simplistic and lack the detailed appearance and geometry needed for high-quality animations. Starting with these simple animated meshes, as shown in Figure 3 (b), we generate an initial visually appealing frame using a depth-to-image FLUX model [Labs 2024]. We then produce 3D tracking videos from the animated meshes, which, when combined with the generated first frame, guide DaS to transform the basic meshes into visually rich and appealing videos.

3.4.3 Camera control. Previous approaches [He et al. 2024b; Wang et al. 2024c] rely on camera or ray embeddings as conditions to control the camera trajectory in video generation. However, these embeddings lack true 3D awareness, leaving the diffusion models to infer the scene's 3D structure and simulate camera movement.

In contrast, DaS significantly enhances 3D awareness by incorporating 3D tracking videos for precise camera control. To generate videos with a specific camera trajectory, as shown in Figure 3 (c), we first estimate the depth map of the initial frame using Depth Pro [Bochkovskii et al. 2024] and convert it into colored 3D points. These points are then projected onto the given camera trajectory, constructing a 3D tracking video that enables DaS to control camera movements with high 3D accuracy.

3.4.4 Motion transfer. As shown in Figure 3 (d), DaS also facilitates creating a new video by transferring motion from an existing source video. First, we estimate the depth map of the source video's first frame and apply the depth-to-image FLUX model [Labs 2024] to repaint the frame into a target appearance guided by text prompts. Then, using SpatialTracker [Xiao et al. 2024b], we generate a 3D tracking video from the source video to serve as control signals. Finally, the DaS model generates the target video by combining the edited first frame with the 3D tracking video.

4 EXPERIMENTS

We conduct experiments on five tasks, including camera control, motion transfer, mesh-to-video generation, and object manipulation to demonstrate the versatility of DaS in controlling the video generation process.

4.1 Camera control

Baseline methods. To evaluate the ability to control camera motions of generated videos, we select two representative methodologies, MotionCtrl [Wang et al. 2024c] and CameraCtrl [He et al. 2024b] as baseline methods, both of which allow camera trajectories as input and use camera or ray embeddings for camera control.
Metrics. To measure the accuracy of the camera trajectories of generated videos, we evaluate the consistency between the estimated

Fig. 4. **Qualitative results of DaS on the camera control task**. We show 4 trajectories (left, right, up, down) with large movements.

camera poses from the generated videos and the input ground-truth camera poses using rotation errors and translation errors. Specifically, for each frame of a generated video, we reconstruct its relative pose given the first frame using SIFT [Ng and Henikoff 2003]. Then, we get the normalized quaternion and translation vectors for the rotation and translation. Finally, we calculate the cosine similarity between the estimated camera poses with the given camera poses.

$$\mathbf{RotErr} = \arccos\left(\frac{1}{T-1}\sum_{i=2}^{T}\langle\,\mathbf{q}_{\text{gen}}^{i}, \mathbf{q}_{\text{gt}}^{i}\,\rangle\right),$$

$$\mathbf{TransErr} = \arccos\left(\frac{1}{T-1}\sum_{i=2}^{T}\langle\,\mathbf{t}_{\text{gen}}^{i}, \mathbf{t}_{\text{gt}}^{i}\,\rangle\right),$$

where T is the number of frames, $\mathbf{q}^{i}$ and $\mathbf{t}^{i}$ are the normalized quaternion and translation vector of the i-th frame, and $\langle\cdot,\cdot\rangle$ means the dot product between two vectors.

Results. We compare against baseline methods on 100 random trajectories from RealEstate10K [Zhou et al. 2018]. But since most of the random trajectories only contain small movements, we further test the models on larger fixed movements (moving left, right, up, down, spiral) as shown in Figure 4. As shown in Table 1, our method outperforms the baseline methods, which demonstrates that our method achieves stable and accurate control of the camera poses of the generated videos. The main reason is that due to the utilization of the 3D tracking videos, our method is fully 3D-aware to enable accurate spatial inference in the video generation process. In comparison, baseline methods [He et al. 2024b; Wang et al. 2024c] only adopt implicit camera or ray embeddings for camera control.

Method	Small Movement		Large Movement	
	TransErr ↓	RotErr ↓	TransErr ↓	RotErr ↓
MotionCtrl	44.23	8.92	67.05	39.86
CameraCtrl	42.31	7.82	66.76	29.70
Ours	**27.85**	**5.97**	**37.17**	**10.40**

Table 1. **Quantitative results on camera control** of MotionCtrl [Wang et al. 2024c], CameraCtrl [He et al. 2024b], and our method. "TransErr" and "RotErr" are the angle differences between the estimated translation and rotation and the ground-truth ones in degree.

4.2 Motion transfer

Baseline methods. We compare DaS with two famous motion transfer methods, TokenFlow [Geyer et al. 2023b] and CCEdit [Feng et al. 2024b]. TokenFlow represents video motions with the feature consistency across different timesteps extracted by a diffusion model. Then, the feature consistency is propagated to several keyframes generated by a text prompt for video generation. For TokenFlow, we adopt the Stable Diffusion 2.1 [Rombach et al. 2022] model for the motion transfer task. CCEdit adopts depth maps as conditions to control the video motion and transfers the motion using a new repainted frame to generate a video.

Metrics. Since all methods generate the transferred videos based on text prompts, we aim to evaluate the alignment between the generated videos and the text prompts, as well as the video coherence, using the CLIP [Radford et al. 2021]. Specifically, for video-text alignment, we extract multiple frames from the video and compare them with the corresponding text prompts by calculating the CLIP score [Hessel et al. 2022] for each frame. This score reflects
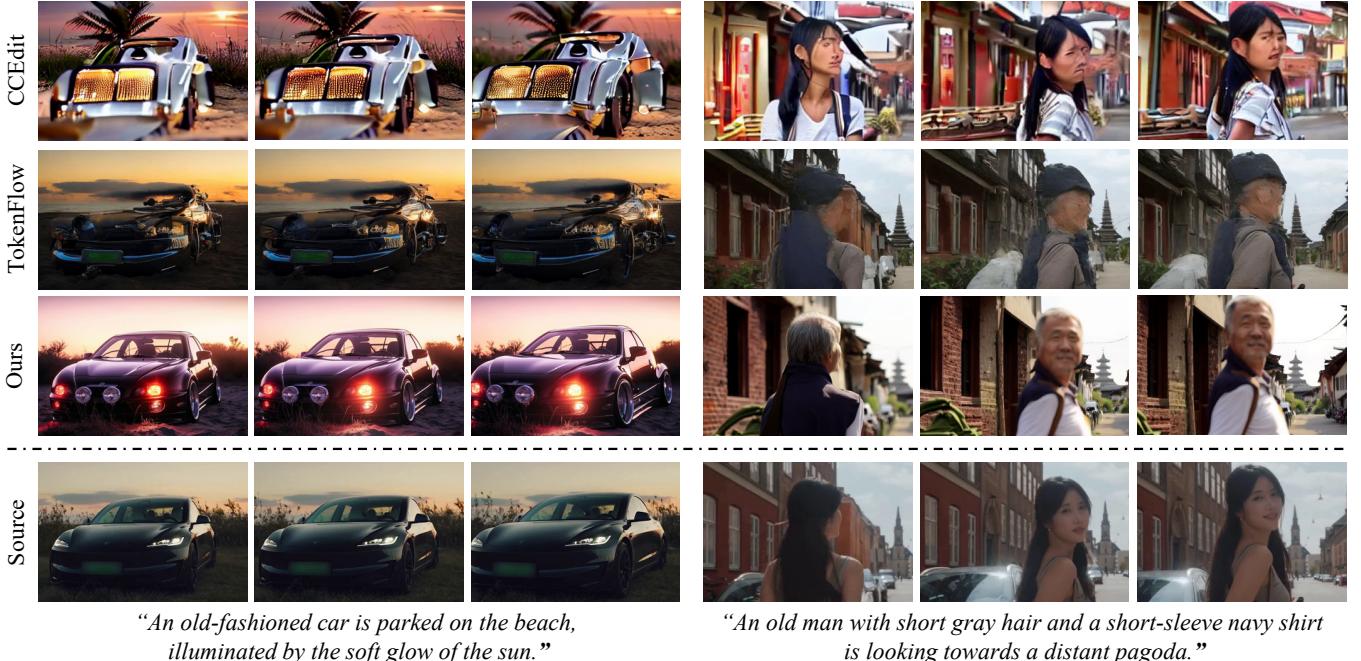
"An old-fashioned car is parked on the beach, illuminated by the soft glow of the sun."

"An old man with short gray hair and a short-sleeve navy shirt is looking towards a distant pagoda."

Fig. 5. **Qualitative comparison on motion transfer** between our method, CCEdit [Feng et al. 2024b], and TokenFlow [Geyer et al. 2023b].

Method	Tex-Ali ↑	Tem-Con ↑
CCEdit	16.9	0.932
Tokenflow	31.9	0.956
Ours	**32.6**	**0.971**

Table 2. **CLIP scores for motion transfer** of CCEdit [Feng et al. 2024b], TokenFlow [Geyer et al. 2023b], and our method. "Text-Ali" is the semantic CLIP consistency between generated videos and the given text prompts. "Tem-Con" is the temporal CLIP consistency between neighboring frames.

Depth	Tracking	#Tracks	PSNR ↑	SSIM ↑	LPIPS ↓	FVD ↓
✓		-	18.08	0.573	0.312	645.1
	✓	900	18.52	0.586	0.337	765.3
	✓	2500	19.17	0.632	0.263	566.4
	✓	4900	**19.27**	**0.658**	**0.261**	**551.3**
	✓	8100	19.11	0.649	0.262	599.0

Table 3. **Analysis of applying different 3D control signals for image to video generation**. We evaluate PSNR, SSIM, LPIPS, and FVD of generated videos on the validation set of the DAVIS and MiraData datasets. "Depth" means using depth maps as the 3D control signals. "Tracking" means using 3D tracking videos as the control signals. #Tracks means the number of 3D points used in the 3D tracking video.

the alignment between image content and textual descriptions. For temporal consistency, we extract normalized CLIP features from adjacent video frames and compute the cosine similarity between the adjacent features.

Results. As shown in Table 2, our method demonstrates outstanding performance in both text alignment and frame consistency, surpassing two baseline methods. Furthermore, Figure 5 presents the qualitative comparison of our method, CCEdit, and TokenFlow. It shows that CCEdit produces frames of low quality and struggles to maintain temporal coherence. TokenFlow produces semantically consistent frames but has difficulty producing coherent videos. In contrast, our method accurately transfers the video motion with strong temporal coherence as shown in Figure 6.

4.3 Animating meshes to videos

Qualitative comparison. We compare our method against a state-of-the-art human image animation method CHAMP [Zhu et al. 2024] on the mesh-to-video task. Champ takes a human image and a motion sequence as input and generates a corresponding human video. The motion sequence is represented by an animated SMPL [Loper

et al. 2023] mesh. We use the same input image but the SMPL mesh for CHAMP and generate the corresponding animation videos for qualitative comparison as shown in Figure 8. We also generate different styles of videos from the same animated 3D meshes as shown in Figure 8. Compared to CHAMP, our method demonstrates better consistency in the 3D structure and texture details of the avatar on different motion sequences and across different styles.

4.4 Object manipulation

Qualitative results. For the object manipulation, we adopt the SAM [Kirillov et al. 2023] and depth estimation models [Bochkovskii et al. 2024; Wang et al. 2024b] to get the object points. Then, we evaluate two kinds of manipulation, i.e. translation and rotation. The results are shown in Figure 9, which demonstrate that DaS achieves accurate object manipulation to produce photorealistic videos with strong multiview consistency for these objects.
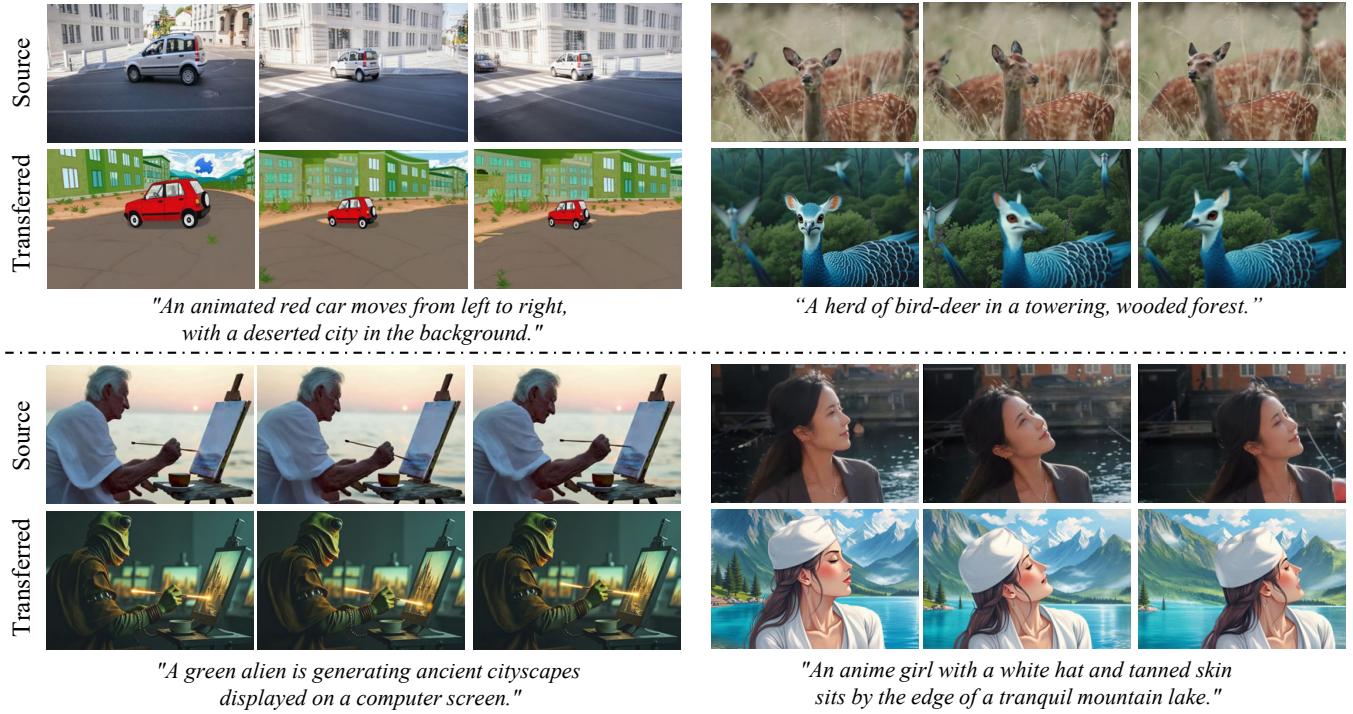
Source

Transferred

"An animated red car moves from left to right, with a deserted city in the background."

"A herd of bird-deer in a towering, wooded forest."

Source

Transferred

"A green alien is generating ancient cityscapes displayed on a computer screen."

"An anime girl with a white hat and tanned skin sits by the edge of a tranquil mountain lake."

Fig. 6. **Qualitative results on motion transfer of our method**.

Style4

Style3
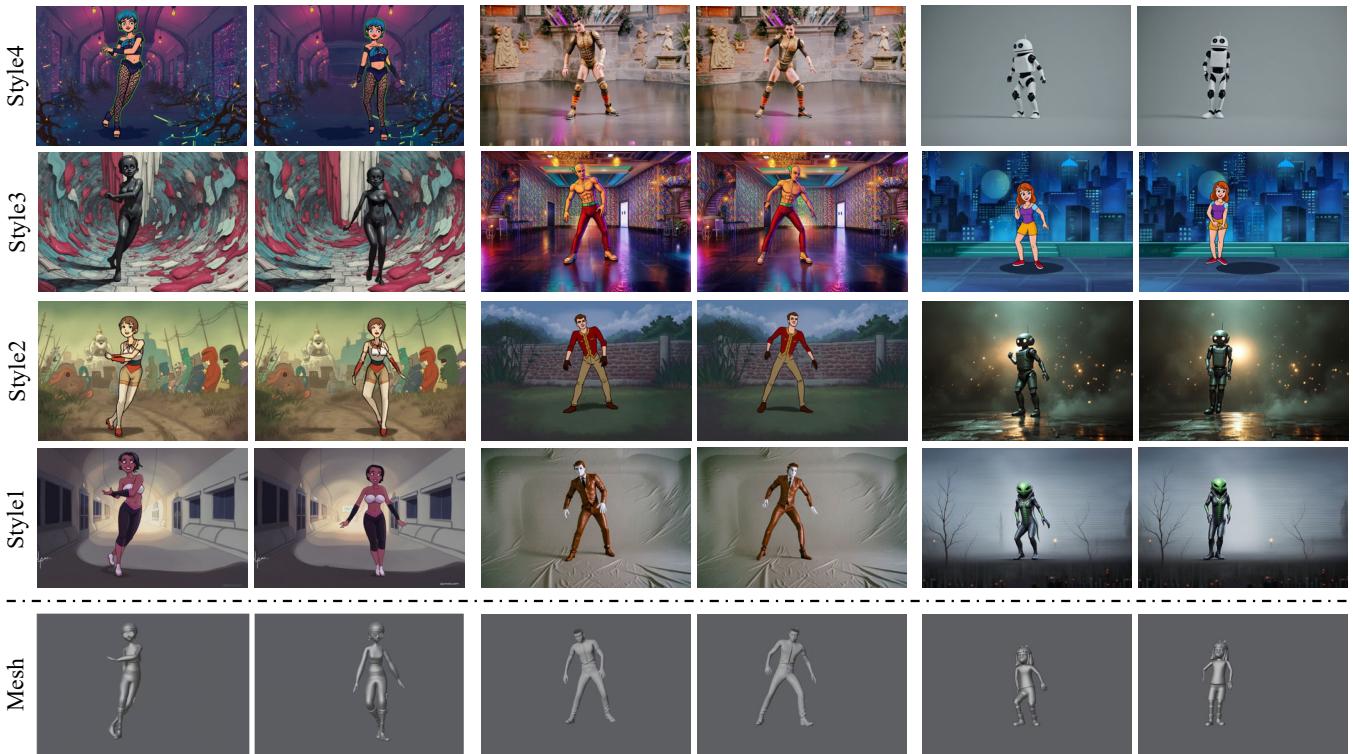
Style2

Style1

Mesh

Fig. 7. **More results of the animating mesh to video generation task**. Our method enables the generation of different styles from the same mesh.
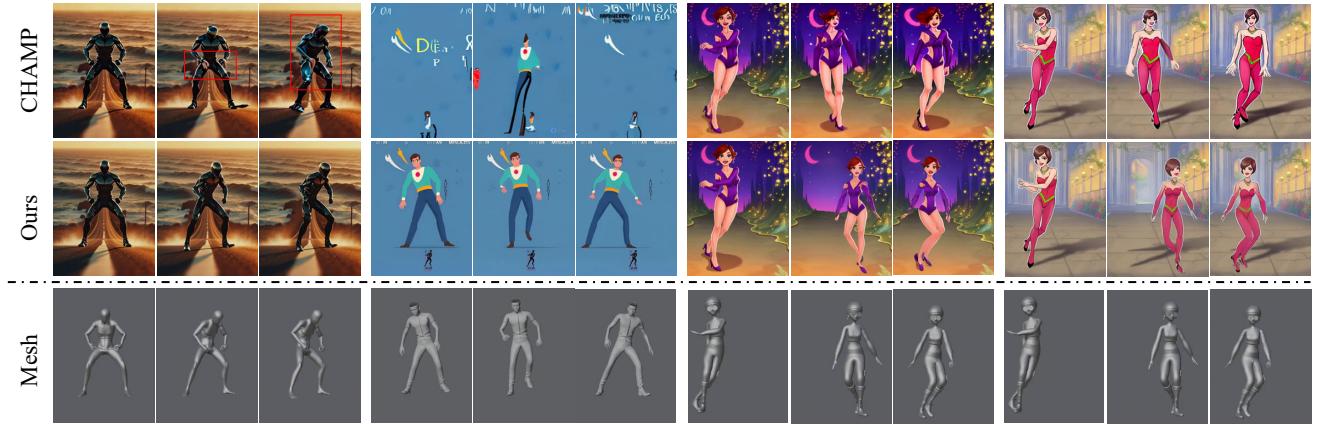
Fig. 8. **Qualitative comparison on the animating mesh to video task** between our method and CHAMP [Zhu et al. 2024].
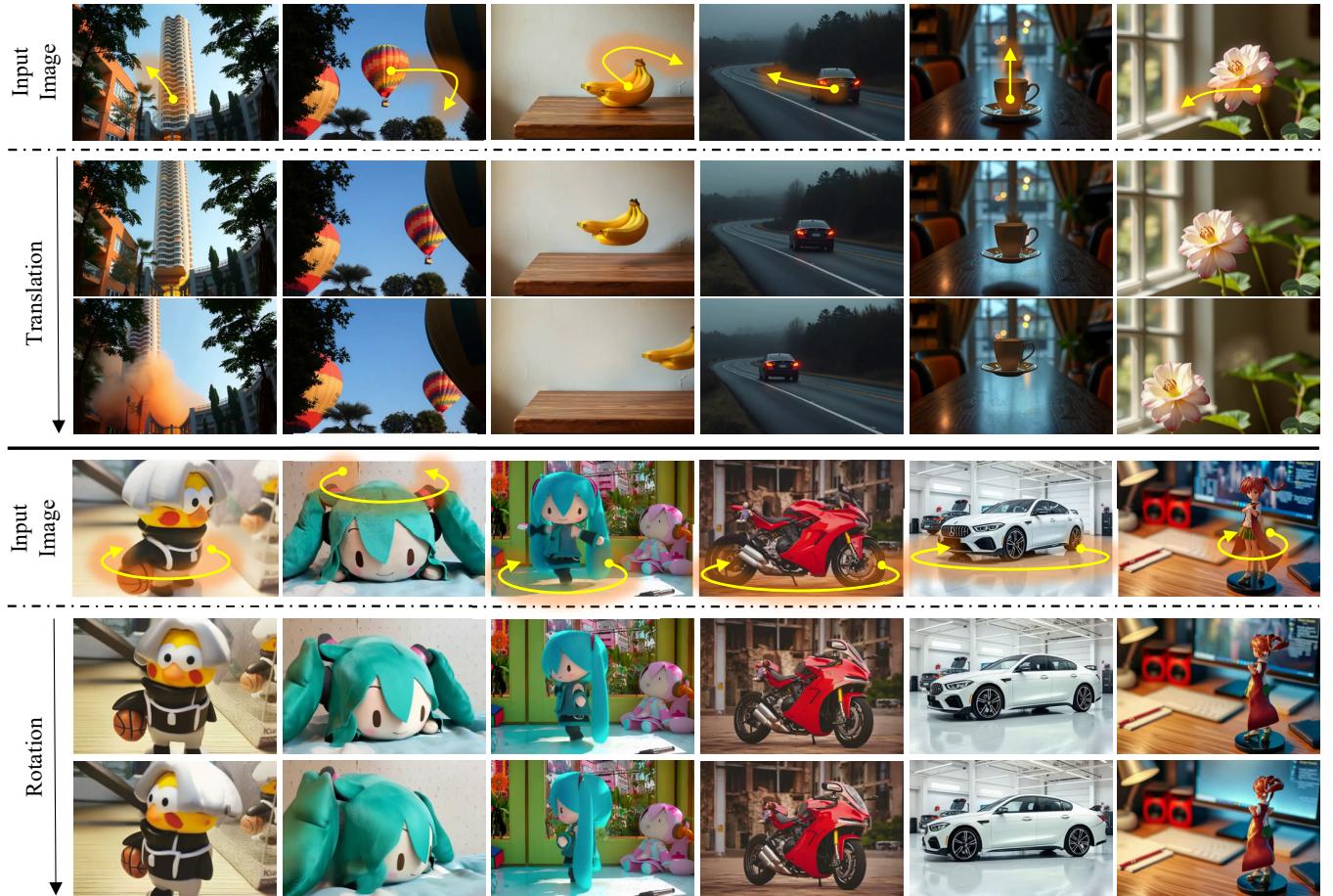

Fig. 9. **Qualitative results of our method on the object manipulation task**. The top part shows the results of translation while the bottom part shows the results of rotating the object.

Fig. 10. **Generated videos using depth maps or 3D tracking videos as control signals.** Our 3D tracking videos provide better quality on the cross-frame consistency for video generation than depth maps.

4.5 Analysis

We conduct analysis on the choice of 3D control signals, i.e. depth maps or 3D tracking videos, and the number of 3D tracking points. To achieve this, we randomly selected 50 videos from the validation split of the DAVIS [Pont-Tuset et al. 2017] and MiraData [Ju et al. 2024] video dataset. We extract the first-frame images as the input image and apply different models to re-generate these videos. To evaluate the quality of the generated videos, we compute PSNR, SSIM [Wang et al. 2004], LPIPS [Zhang et al. 2018], and FVD [Unterthiner et al. 2019] between the generated videos and the ground-truth videos.

4.5.1 Depth maps vs. 3D tracking videos. To illustrate the effectiveness of our 3D tracking videos, we compare DaS with a baseline using depth maps as conditions instead of 3D tracking videos. Specifically, the baseline adopts the same architecture as DaS but replaces the 3D tracking video with a depth map video. We adopt the Depth Pro [Bochkovskii et al. 2024] to generate the video depth video for this baseline method. As shown in Table 3, our model outperforms this baseline in all metrics, demonstrating that the 3D tracking videos provide a better signal for the diffusion model to recover groud-truth videos than the depth map conditions. Figure 10 shows the generated videos, which demonstrate that our method produces more consistent videos with the ground truth. The main reason is that the 3D tracking videos effectively associate different frames of a video while the depth maps only provide some cues of the scene structures without constraining the motion of the video.

4.5.2 Point density. In Table 3, we further present an ablation study with varying numbers of 3D tracking points as control signals. The number of 3D tracking points ranges from 900 (30×30) to 8100 (90×90). Though the generated videos with 4900 tracking points perform slightly better than the other ones, the visual qualities of 2500, 4900, and 8100 tracking points are very similar to each other. Since tracking too many points with SpatialTracker [Xiao et al. 2024b] would be slow, we choose 4900 as our default setting in all our other experiments using 3D point tracking.

4.5.3 Runtime. In the inference stage, we employ the DDIM [Song et al. 2020] sampler with 50 steps, classifier-free guidance of magnitude 7.0, which costs about 2.5 minutes to generate 49 frames on a H800 GPU at a resolution of 480×720.
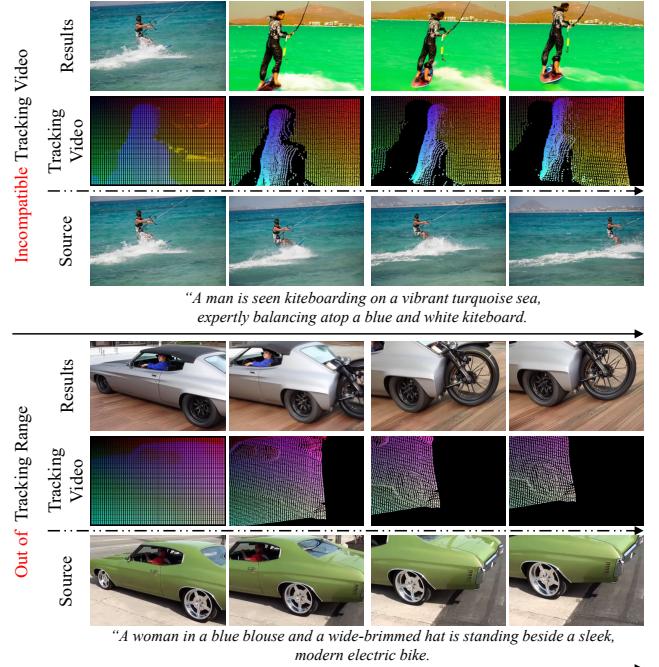

Fig. 11. **Failure cases**. (Top) Incompatible tracking video. When a tracking video that does not correspond to the structures of the input image is provided, DaS will generate a video with a scene transition to a compatible new scene. (Bottom) Out of tracking range. For regions without 3D tracking points, the tracking video fails to constrain these regions and DaS may generate some uncontrolled content.

5 LIMITATIONS AND CONCLUSIONS

Limitations and future works. Though DaS achieves control over the video generation process in most cases, it still suffers from multiple failure cases mainly caused by incorrect 3Dtracking videos. The first failure case is that the input image should be compatible with the 3D tracking videos. Otherwise, the generated videos would be implausible as shown in Figure 11 (top). Another failure case is that for regions without 3D tracking points, the generated contents may be out-of-control and produce some unnatural results (Figure 11 (bottom)). For future works, we currently rely on provided animated meshes or existing videos to get high-quality 3D tracking videos and a promising direction is to learn to generate these 3D tracking videos with a new diffusion model.

Conclusions. In this paper, we introduce Diffusion as Shader (DaS) for controllable video generation. The key idea of DaS is to adopt the 3D tracking videos as 3D control signals for video generation. The 3D tracking videos are constructed from colored dynamic 3D points which represent the underlying 3D motion of the video. Then, diffusion models are applied to generate a video following the motion of the 3D tracking video. We demonstrate that the 3D tracking videos not only improve the temporal consistency of the generated videos but also enable versatile control of the video content, including mesh-to-video generation, camera control, motion transfer, and object manipulation.

REFERENCES

Sherwin Bahmani, Ivan Skorokhodov, Guocheng Qian, Aliaksandr Siarohin, Willi Menapace, Andrea Tagliasacchi, David B Lindell, and Sergey Tulyakov. 2024. AC3D: Analyzing and Improving 3D Camera Control in Video Diffusion Transformers. *arXiv preprint arXiv:2411.18673* (2024).

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

Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R Richter, and Vladlen Koltun. 2024. Depth pro: Sharp monocular metric depth in less than a second. *arXiv preprint arXiv:2410.02073* (2024).

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. 2024. Video generation models as world simulators. https://openai.com/research/video-generation-models-as-world-simulators

Shengqu Cai, Duygu Ceylan, Matheus Gadelha, Chun-Hao Paul Huang, Tuanfeng Yang Wang, and Gordon Wetzstein. 2024. Generative rendering: Controllable 4d-guided video generation with 2d diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7611–7620.

Tianshi Cao, Karsten Kreis, Sanja Fidler, Nicholas Sharp, and Kangxue Yin. 2023. Texfusion: Synthesizing 3d textures with text-guided image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4169–4181.

Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. 2023b. Videocrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512* (2023).

Haoxin Chen, Yong Zhang, Xiaodong Cun, Menghan Xia, Xintao Wang, Chao Weng, and Ying Shan. 2024b. Videocrafter2: Overcoming data limitations for high-quality video diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7310–7320.

Junsong Chen, Chongjian Ge, Enze Xie, Yue Wu, Lewei Yao, Xiaozhe Ren, Zhongdao Wang, Ping Luo, Huchuan Lu, and Zhenguo Li. 2024a. PIXART-Sigma: Weak-to-Strong Training of Diffusion Transformer for 4K Text-to-Image Generation. In *European Conference on Computer Vision*. Springer, 74–91.

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

Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. 2023. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7346–7356.

Ruoyu Feng, Wenming Weng, Yanhui Wang, Yuhui Yuan, Jianmin Bao, Chong Luo, Zhibo Chen, and Baining Guo. 2024b. CCEdit: Creative and Controllable Video Editing via Diffusion Models. arXiv:2309.16496 [cs.CV] https://arxiv.org/abs/2309.16496

Wanquan Feng, Tianhao Qi, Jiawei Liu, Mingzhen Sun, Pengqi Tu, Tianxiang Ma, Fei Dai, Songtao Zhao, Siyu Zhou, and Qian He. 2024a. I2VControl: Disentangled and Unified Video Motion Synthesis Control. *arXiv preprint arXiv:2411.17765* (2024).

Daniel Geng, Charles Herrmann, Junhwa Hur, Forrester Cole, Serena Zhang, Tobias Pfaff, Tatiana Lopez-Guevara, Carl Doersch, Yusuf Aytar, Michael Rubinstein, et al. 2024. Motion Prompting: Controlling Video Generation with Motion Trajectories. *arXiv preprint arXiv:2412.02700* (2024).

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. 2023a. Tokenflow: Consistent diffusion features for consistent video editing. *arXiv preprint arXiv:2307.10373* (2023).

Michal Geyer, Omer Bar-Tal, Shai Bagon, and Tali Dekel. 2023b. TokenFlow: Consistent Diffusion Features for Consistent Video Editing. arXiv:2307.10373 [cs.CV] https://arxiv.org/abs/2307.10373

Yuwei Guo, Ceyuan Yang, Anyi Rao, Maneesh Agrawala, Dahua Lin, and Bo Dai. 2024. Sparsectrl: Adding sparse controls to text-to-video diffusion models. In *ECCV*. 330–348.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. 2023. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725* (2023).

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

Xuanhua He, Quande Liu, Shengju Qian, Xin Wang, Tao Hu, Ke Cao, Keyu Yan, and Jie Zhang. 2024a. Id-animator: Zero-shot identity-preserving human video generation. *arXiv preprint arXiv:2404.15275* (2024).

Yingqing He, Tianyu Yang, Yong Zhang, Ying Shan, and Qifeng Chen. 2022. Latent video diffusion models for high-fidelity long video generation. *arXiv preprint arXiv:2211.13221* (2022).

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. 2022. CLIPScore: A Reference-free Evaluation Metric for Image Captioning. arXiv:2104.08718 [cs.CV] https://arxiv.org/abs/2104.08718

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *NeurIPS* (2020).

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. 2022. Video diffusion models. *Advances in Neural Information Processing Systems* 35 (2022), 8633–8646.

Hsin-Ping Huang, Yu-Chuan Su, Deqing Sun, Lu Jiang, Xuhui Jia, Yukun Zhu, and Ming-Hsuan Yang. 2023. Fine-grained controllable video generation via object appearance and context. *arXiv preprint arXiv:2312.02919* (2023).

Yash Jain, Anshul Nasery, Vibhav Vineet, and Harkirat Behl. 2024. Peekaboo: Interactive video generation via masked-diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8079–8088.

Hyeonho Jeong, Chun-Hao Paul Huang, Jong Chul Ye, Niloy Mitra, and Duygu Ceylan. 2024. Track4Gen: Teaching Video Diffusion Models to Track Points Improves Video Generation. *arXiv preprint arXiv:2412.06016* (2024).

Xuan Ju, Yiming Gao, Zhaoyang Zhang, Ziyang Yuan, Xintao Wang, Ailing Zeng, Yu Xiong, Qiang Xu, and Ying Shan. 2024. MiraData: A Large-Scale Video Dataset with Long Durations and Structured Captions. arXiv:2407.06358 [cs.CV] https://arxiv.org/abs/2407.06358

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4015–4026.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, et al. 2024. Hunyuan-Video: A Systematic Framework For Large Video Generative Models. *arXiv preprint arXiv:2412.03603* (2024).

Mathis Koroglu, Hugo Caselles-Dupré, Guillaume Jeanneret Sanmiguel, and Matthieu Cord. 2024. OnlyFlow: Optical Flow based Motion Conditioning for Video Diffusion Models. *arXiv preprint arXiv:2411.10501* (2024).

Kuaishou. 2024. Keling. https://kling.kuaishou.com/

Black Forest Labs. 2024. FLUX. https://github.com/black-forest-labs/flux

Guojun Lei, Chi Wang, Hong Li, Rong Zhang, Yikai Wang, and Weiwei Xu. 2024. AnimateAnything: Consistent and Controllable Animation for Video Generation. *arXiv preprint arXiv:2411.10836* (2024).

Yaowei Li, Xintao Wang, Zhaoyang Zhang, Zhouxia Wang, Ziyang Yuan, Liangbin Xie, Yuexian Zou, and Ying Shan. 2024. Image conductor: Precision control for interactive video synthesis. *arXiv preprint arXiv:2406.15339* (2024).

Bin Lin, Yunyang Ge, Xinhua Cheng, Zongjian Li, Bin Zhu, Shaodong Wang, Xianyi He, Yang Ye, Shenghai Yuan, Liuhan Chen, et al. 2024. Open-Sora Plan: Open-Source Large Video Generation Model. *arXiv preprint arXiv:2412.00131* (2024).

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. 2023. SMPL: A skinned multi-person linear model. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 851–866.

Jiahao Lu, Tianyu Huang, Peng Li, Zhiyang Dou, Cheng Lin, Zhiming Cui, Zhen Dong, Sai-Kit Yeung, Wenping Wang, and Yuan Liu. 2024. Align3R: Aligned Monocular Depth Estimation for Dynamic Videos. *arXiv preprint arXiv:2412.03079* (2024).

Wan-Duo Kurt Ma, John P Lewis, and W Bastiaan Kleijn. 2024b. Trailblazer: Trajectory control for diffusion-based video generation. In *SIGGRAPH Asia*.

Yue Ma, Yingqing He, Hongfa Wang, Andong Wang, Chenyang Qi, Chengfei Cai, Xiu Li, Zhifeng Li, Heung-Yeung Shum, Wei Liu, et al. 2024a. Follow-your-click: Open-domain regional image animation via short prompts. *arXiv preprint arXiv:2403.08268* (2024).

Tuna Han Salih Meral, Hidir Yesiltepe, Connor Dunlop, and Pinar Yanardag. 2024. MotionFlow: Attention-Driven Motion Transfer in Video Diffusion Models. *arXiv preprint arXiv:2412.05275* (2024).

Chong Mou, Mingdeng Cao, Xintao Wang, Zhaoyang Zhang, Ying Shan, and Jian Zhang. 2024. ReVideo: Remake a Video with Motion and Content Control. *arXiv preprint arXiv:2405.13865* (2024).

Koichi Namekata, Sherwin Bahmani, Ziyi Wu, Yash Kant, Igor Gilitschenski, and David B Lindell. 2024. Sg-i2v: Self-guided trajectory control in image-to-video generation. *arXiv preprint arXiv:2411.04989* (2024).

Pauline C Ng and Steven Henikoff. 2003. SIFT: Predicting amino acid changes that affect protein function. *Nucleic acids research* 31, 13 (2003), 3812–3814.

Muyao Niu, Xiaodong Cun, Xintao Wang, Yong Zhang, Ying Shan, and Yinqiang Zheng. 2024. Mofa-video: Controllable image animation via generative motion field adaptions in frozen image-to-video diffusion model. In *ECCV*.

Geon Yeong Park, Hyeonho Jeong, Sang Wan Lee, and Jong Chul Ye. 2024. Spectral motion alignment for video motion transfer using diffusion models. *arXiv preprint arXiv:2403.15249* (2024).

William Peebles and Saining Xie. 2023a. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4195–4205.

William Peebles and Saining Xie. 2023b. Scalable Diffusion Models with Transformers. arXiv:2212.09748 [cs.CV] https://arxiv.org/abs/2212.09748

Adam Polyak, Amit Zohar, Andrew Brown, Andros Tjandra, Animesh Sinha, Ann Lee, Apoorv Vyas, Bowen Shi, Chih-Yao Ma, Ching-Yao Chuang, et al. 2024. Movie gen: A cast of media foundation models. *arXiv preprint arXiv:2410.13720* (2024).

Alexander Pondaven, Aliaksandr Siarohin, Sergey Tulyakov, Philip Torr, and Fabio Pizzati. 2024. Video Motion Transfer with Diffusion Transformers. *arXiv preprint arXiv:2412.07776* (2024).

Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alexander Sorkine-Hornung, and Luc Van Gool. 2017. The 2017 DAVIS Challenge on Video Object Segmentation. *arXiv:1704.00675* (2017).

Haonan Qiu, Zhaoxi Chen, Zhouxia Wang, Yingqing He, Menghan Xia, and Ziwei Liu. 2024. Freetraj: Tuning-free trajectory control in video diffusion models. *arXiv preprint arXiv:2406.16863* (2024).

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020 [cs.CV] https://arxiv.org/abs/2103.00020

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. 2020. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence* 44, 3 (2020), 1623–1637.

Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. 2023. Texture: Text-guided texturing of 3d shapes. In *ACM SIGGRAPH 2023 conference proceedings*. 1–11.

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

Xiaoyu Shi, Zhaoyang Huang, Fu-Yun Wang, Weikang Bian, Dasong Li, Yi Zhang, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, et al. 2024. Motion-i2v: Consistent and controllable image-to-video generation with explicit motion modeling. In *SIGGRAPH*.

Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. 2021. Light field networks: Neural scene representations with single-evaluation rendering. *Advances in Neural Information Processing Systems* 34 (2021), 19313–19325.

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

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

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. 2019. Towards Accurate Generative Models of Video: A New Metric & Challenges. arXiv:1812.01717 [cs.CV] https://arxiv.org/abs/1812.01717

Jiawei Wang, Yuchen Zhang, Jiaxin Zou, Yan Zeng, Guoqiang Wei, Liping Yuan, and Hang Li. 2024f. Boximator: Generating rich and controllable motions for video synthesis. *arXiv preprint arXiv:2402.01566* (2024).

Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. 2024b. MoGe: Unlocking Accurate Monocular Geometry Estimation for Open-Domain Images with Optimal Training Supervision. arXiv:2410.19115 [cs.CV] https://arxiv.org/abs/2410.19115

Tianfu Wang, Menelaos Kanakis, Konrad Schindler, Luc Van Gool, and Anton Obukhov. 2023. Breathing new life into 3d assets with generative repainting. *arXiv preprint arXiv:2309.08523* (2023).

Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. 2024d. Videocomposer: Compositional video synthesis with motion controllability. *Advances in Neural Information Processing Systems* 36 (2024).

Yuelei Wang, Jian Zhang, Pengtao Jiang, Hao Zhang, Jinwei Chen, and Bo Li. 2024e. CPA: Camera-pose-awareness Diffusion Transformer for Video Generation. *arXiv preprint arXiv:2412.01429* (2024).

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861

Zhouxia Wang, Yushi Lan, Shangchen Zhou, and Chen Change Loy. 2024a. ObjCtrl-2.5 D: Training-free Object Control with Camera Poses. *arXiv preprint arXiv:2412.07721* (2024).

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

Yuxi Xiao, Qianqian Wang, Shangzhan Zhang, Nan Xue, Sida Peng, Yujun Shen, and Xiaowei Zhou. 2024b. SpatialTracker: Tracking Any 2D Pixels in 3D Space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20406–20417.

Zeqi Xiao, Wenqi Ouyang, Yifan Zhou, Shuai Yang, Lei Yang, Jianlou Si, and Xingang Pan. 2024a. Trajectory Attention for Fine-grained Video Motion Control. *arXiv preprint arXiv:2411.19324* (2024).

Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Gongye Liu, Xintao Wang, Ying Shan, and Tien-Tsin Wong. 2024. Dynamicrafter: Animating open-domain images with video diffusion priors. In *European Conference on Computer Vision*. Springer, 399–417.

Shiyuan Yang, Liang Hou, Haibin Huang, Chongyang Ma, Pengfei Wan, Di Zhang, Xiaodong Chen, and Jing Liao. 2024a. Direct-a-video: Customized video generation with user-directed camera movement and object motion. In *ACM SIGGRAPH 2024 Conference Papers*. 1–12.

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

Danah Yatim, Rafail Fridman, Omer Bar-Tal, Yoni Kasten, and Tali Dekel. 2024. Space-time diffusion features for zero-shot text-driven motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8466–8476.

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

Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. 2024. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *arXiv preprint arXiv:2409.02048* (2024).

Shenghai Yuan, Jinfa Huang, Xianyi He, Yunyuan Ge, Yujun Shi, Liuhan Chen, Jiebo Luo, and Li Yuan. 2024. Identity-Preserving Text-to-Video Generation by Frequency Decomposition. *arXiv preprint arXiv:2411.17440* (2024).

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *ICCV*. 3836–3847.

Qihang Zhang, Shuangfei Zhai, Miguel Angel Bautista, Kevin Miao, Alexander Toshev, Joshua Susskind, and Jiatao Gu. 2024. World-consistent Video Diffusion with Explicit 3D Modeling. *arXiv preprint arXiv:2412.01821* (2024).

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. arXiv:1801.03924 [cs.CV] https://arxiv.org/abs/1801.03924

Tingyang Zhang, Chen Wang, Zhiyang Dou, Jiahui Lei Qingzhe Gao, Baoquan Chen, and Lingjie Liu. 2025. ProTracker: Probabilistic Integration for Robust and Accurate Point Tracking. *arXiv preprint arxiv:2501.03220* (2025).

Guangcong Zheng, Teng Li, Rui Jiang, Yehao Lu, Tao Wu, and Xi Li. 2024a. CamI2V: Camera-Controlled Image-to-Video Diffusion Model. *arXiv preprint arXiv:2410.15957* (2024).

Zangwei Zheng, Xiangyu Peng, Tianji Yang, Chenhui Shen, Shenggui Li, Hongxin Liu, Yukun Zhou, Tianyi Li, and Yang You. 2024b. *Open-Sora: Democratizing Efficient Video Production for All*. https://github.com/hpcaitech/Open-Sora

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. 2018. Stereo Magnification: Learning View Synthesis using Multiplane Images. In *SIGGRAPH*.

Shenhao Zhu, Junming Leo Chen, Zuozhuo Dai, Yinghui Xu, Xun Cao, Yao Yao, Hao Zhu, and Siyu Zhu. 2024. Champ: Controllable and Consistent Human Image Animation with 3D Parametric Guidance. arXiv:2403.14781 [cs.CV]