TRAINING-FREE CAMERA CONTROL FOR VIDEO GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a training-free and robust solution to offer camera movement control for off-the-shelf video diffusion models. Unlike previous work, our method does not require any supervised finetuning on camera-annotated datasets or self-supervised training via data augmentation. Instead, it can be plugged and played with most pretrained video diffusion models and generate camera controllable videos with a single image or text prompt as input. The inspiration of our work comes from the layout prior that intermediate latents hold towards generated results, thus rearranging noisy pixels in them will make output content reallocated as well. As camera move could also be seen as a kind of pixel rearrangement caused by perspective change, videos could be reorganized following specific camera motion if their noisy latents change accordingly. Established on this, we propose our method **CamTrol**, which enables robust camera control for video diffusion models. It is achieved by a two-stage process. First, we model image layout rearrangement through explicit camera movement in 3D point cloud space. Second, we generate videos with camera motion using layout prior of noisy latents formed by a series of rearranged images. Extensive experiments have demonstrated its superior performance in both video generation quality and camera motion alignment compared with other finetuned methods. Furthermore, we show the capacity of CamTrol generalizing to various base models, as well as its impressive applications in scalable motion control, dealing with complicated trajectories and unsupervised 3D video generation. Videos available at **anonymous demo page**. (If not accessible, please refer to the supplementary materials for the same demo.)

1 INTRODUCTION

As a more appealing and content-richer modality, videos differ from images by including an extra temporal dimension. This temporal aspect provides increased versatility for depicting diverse and dynamic movements, which can be decomposed into object motion, background transitions and perspective changes. Recent years have witnessed the rapid development and splendid breakthrough of video generation with text prompt or images as input instructions (Li et al., 2023; Hong et al., 2022; Ho et al., 2022; Luo et al., 2023; Zeng et al., 2024; Blattmann et al., 2023a; Brooks et al., 2024; Ge et al., 2023; Fei et al., 2023), and demonstrated the inestimable potential of diffusion models to synthesis realistic videos. While these video generation models have made progress in generating videos with highly dynamic objects and backgrounds (Zeng et al., 2024; Blattmann et al., 2023a; Li et al., 2023), most of them fail to provide camera control for the generated videos.

The difficulty of controlling camera trajectory in videos primarily arises from two aspects. The initial challenge lies in the inadequacy of annotated data. Most video annotations lack of descriptions, especially precise descriptions of video's camera movements. As a result, video generation models trained on these data often fail to interpret text prompts related to camera motions and generate correct outputs. One solution to mitigate the data insufficiency problem is to mimic videos with camera movements through simple data augmentation (Yang et al., 2024a). However, these methods could only handle simple camera motions like *zoom* or *truck*, and have trouble in dealing with more complicated ones. The second challenge lies in the effort of additional finetuning required for controlling camera movements. As camera trajectories could be sophisticated, they sometimes cannot be accurately elaborated using naïve text prompts alone. Common solutions (Wang et al., 2023; He et al., 2024) proposed to embed camera parameters into diffusion models through learnable encoders and perform extensive finetuning on large-scale datasets with detailed camera trajectories. However, such datasets like RealEstate10k (Zhou et al., 2018) and MVImageNet (Yu et al., 2023) are intensively limited in scale and diversity due to the difficulty associated with data collection, in this way, these finetuning methods demand substantial resources but exhibit limited generalizability to other types of data. *Lack of annotations and heavy finetuning effort make camera control a challenging task in video generations*.

In this work, we attempt to address these issues through a training-free solution to offer camera control for off-the-shelf video diffusion models. We begin by introducing two core observations underpin that video diffusion models can achieve camera movement control in a *training-free* manner. First, we find that base video models could produce results with rough camera moves by integrating specific camera-related text into input prompts, such as *camera zooms in* or *camera pans right*. This simple implementation, though not very accurate and always leads to static or wrong motions, shows the natural prior knowledge learnt by pretrained models about following different camera trajectories. The other observation is the effectiveness video models have exhibited in adapting to 3D generation tasks. Recent works (Voleti et al., 2024; Melas-Kyriazi et al., 2024; Shi et al., 2023) find that leveraging pretrained video models as initialization helps drastically improve the performance of multi-view generations, demonstrating their strong ability of handling perspective change. The two crucial observations reveals the hidden power of video models for camera motion control, thus, we seek to find a way to evoke this innate ability, as it already exists in the model itself.

We propose **CamTrol**, which offers camera control for off-the-shelf video diffusion models in a training-free but robust manner. CamTrol is inspired by the layout prior that noisy latents hold towards generation results: As pixels in noisy latents change their positions, corresponding rearrangement will also occur to the output and leads to layout modification. Considering camera moves could also be seen as a kind of layout rearrangement, this prior can serve as an efficacious hint providing video model with information of specific camera motions. Specifically, CamTrol consists of a two-stage procedure. In stage I, explicit camera movements are modeled in 3D point cloud representations and produce a series of rendered images indicating specific camera movements. In stage II, layout prior of noisy latents are utilized to guide video generations with camera movements. Compared with previous works, CamTrol require no any additional finetuning utilizing camera annotated datasets, nor does it need self-supervised training based on data augmentation. Extensive experiments have demonstrated its superior performance in both video generation quality and camera motion alignment against other finetuned methods. Furthermore, we show the capacity of CamTrol generalizing to various base models, as well as its impressive applications in scalable motion control, dealing with complicated trajectories and unsupervised 3D video generation.

2 Related Work

Camera Control for Video Generation While methods aim for controlling video foundation models constantly emerge (Ma et al., 2024; Liu et al., 2023; Feng et al., 2023), there are few works explore how to manipulate camera motion of generated videos. Earlier work (Hao et al., 2018) controls motion trajectory via warping image through densified sparse flow and pixel fusion, similar ideas also appear later in Chen et al. (2023) and Yin et al. (2023). Besides utilizing optical flow, two main techniques for implementing video camera control are via self-supervised augmentation or additional finetuning. Yang et al. (2024a) disentangles object motion with camera movement and incorporates extra layers to embed camera motions, where model is trained in a self-supervised manner by augmenting input videos to stimulate simple camera movements. He et al. (2024) and Wang et al. (2023) train an additional camera encoder and integrate the output into temporal attention layers of U-Net. Guo et al. (2023) learns new motion pattern via LoRA (Hu et al., 2021) and finetuning with multiple reference videos.

Noise Prior of Latents in Diffusion Model One of the most natural advantages of diffusion model comes from its pixel-wise noisy latents formed during denoising process. These latents hold strong causuality towards output and directly determine what the result looks like, meanwhile have robust error-resilience as they are perturbed by Gaussian noises across different scales. Numerous work have exploited the convenience of this noise prior to attain controllable generation, such as image-to-image translation (Meng et al., 2021), pixel-level manipulation (Nichol et al., 2022), image inpainting (Lugmayr et al., 2022) and semantic editing (Choi et al., 2021; Hou et al., 2024).

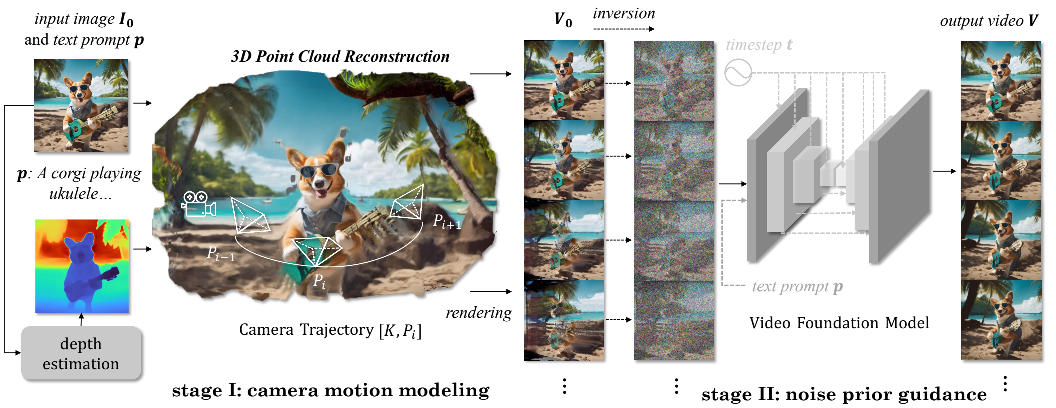


Figure 1: **Pipeline of CamTrol**. In stage I, camera movements are modeled through explicit 3D point cloud . In stage II, layout prior of noisy latents are utilized to guide video generation.

Recent research has shown that even sampled from Gaussian distribution, the initial noise of diffusion process still have significantly influence to the layout of generated contents (Mao et al., 2023). In other work, noise prior is used to guarantee temporal consistency among video frames (Luo et al., 2023), or to trade-off between fidelity and diversity of image editing (Kim et al., 2022).

Video Model for 3D Generation Similar to how most video generation models using the groundwork laid by image foundation models (Blattmann et al., 2023b; Esser et al., 2023; Singer et al., 2022; Wu et al., 2023), training of 3D generation model also relies heavily on pretrained 2D video models (Voleti et al., 2024; Melas-Kyriazi et al., 2024; Shi et al., 2023; Chen et al., 2024; Han et al., 2024). These methods either finetunes with rendered videos directly (Blattmann et al., 2023a; Chen et al., 2024; Melas-Kyriazi et al., 2024; Han et al., 2024), or adds camera embedding for each view as extra condition (Voleti et al., 2024; Shi et al., 2023). Video foundation models have shown to be particularly beneficial in generating consistent multi-view rendering of 3D objects, demonstrates their inherent abundant prior knowledge for handling camera pose change.

3 Training-free Camera Control for Video Generation

CamTrol takes two stages to evoke the innate camera control ability hidden in foundation models. In Sec. 3.1, we will describe how to model explicit camera movement for video generation. In Sec. 3.2, we will elaborate on video's motion control with the guidance of noise layout prior.

3.1 Camera Motion Modeling

To evoke pretrained video diffusion model's ability of dealing with camera perspective changes, hints of camera motion should be injected to diffusion model in a proper way. While simply concatenating camera trajectories with text prompt is incomprehensible for original model, previous works (Wang et al., 2023; He et al., 2024) introduce additional embedder to encode camera parameters and finetune with limited annotated data (Zhou et al., 2018; Yu et al., 2023), which are data-hungry yet lack of generalization ability. Other method (Yang et al., 2024a) constructs camera motions by self-supervised augmentations, but could only handle a few easy camera controls. Thus, we seek a more efficient and robust way to guide the model towards camera controllable.

Considering perspective change of video is originally caused by camera movements in 3D space, we resort to 3D representation for providing generation models with explicit motion hints. Specifically, we choose point cloud as the intermediate representation, in which space we can expediently manipulate camera poses for simulating diverse movements. One extra benefit that point cloud brings is its data-efficiency: By utilizing inpainting techniques, only one single input image is required for the whole point cloud reconstruction, this sidesteps the effort of large-scale finetuning.

Point Cloud Initialization We start by lifting the pixels in input image plane to 3D point cloud representations. In practice, the input image can be either user-defined or created by image generators like Stable Diffusion (Rombach et al., 2022). Given an input image $\mathbf{I}_0 \in \mathbb{R}^{3\times H\times W}$, we first estimate its depth map $\mathbf{D}_0$ using off-the-shelf monocular depth estimator ZeoDepth (Bhat et al., 2023). By combining image and its depth map, point cloud $\mathcal{P}_0$ can be initialized as:

$$\mathcal{P}_0 = \phi([\mathbf{I}_0, \mathbf{D}_0], \mathbf{K}, \mathbf{P}_0), \tag{1}$$

where ϕ denotes the mapping function from RGBD to 3D point cloud, $\mathbf{K}$ and $\mathbf{P}_0$ represent camera's intrinsic and extrinsic matrices set by convention (Chung et al., 2023) as they're usually intractable.

Camera Trajectories To get consistent images from multiple views, we set camera motion as a trajectory of extrinsic matrix $\{\mathbf{P}_1, ..., \mathbf{P}_{N-1}\}$, each including a rotation matrix and translation matrix representing camera's pose and position. At each step i, we project the point cloud back to camera plane using function ψ and get a rendered image with perspective change: $\mathbf{I}_i = \psi(\mathcal{P}_i, \mathbf{K}, \mathbf{P}_i)$. By calculating extrinsic matrices of corresponding movement, we obtain a series of camera motions including zoom, tilt, pan, pedestal, truck, roll and rotate, enabling flexible camera movements. Detailed definitions of these movements are elaborated in Appendix B. By combining basic trajectories, hybrid camera movements can be attained and produce videos with cinematic charm. What's more, benefit from explicit camera motion modeling, our method could support trajectories with precise extrinsics, which means it can generate videos with any complicated camera motion.

Multi-view Consistency When perspective changes, there can be vacancies appear as some areas are unoccupied within the point cloud. To get more reasonable results, we employ image inpainting model (Rombach et al., 2022) to fill up the holes for new renderings, with a mask distinguishing the known points from nonexistent ones. This operation guarantees coherence between known views and novel views in 2D space. After inpainting, image is lifted again onto 3D space and gradually complete the whole point cloud representation. During this process, points between adjacent views may become misaligned since depth estimator only estimates relative depth, further leads to inconsistency in both 3D point cloud and rendered images. To avoid this situation, we adopt depth coefficient optimization (Chung et al., 2023) at each step of the camera movement, formed as:

$$d_i = \operatorname*{argmin}_{d} \left(\sum_{M} \left\| \phi([\tilde{\mathbf{I}}_i, d\tilde{\mathbf{D}}_i], \mathbf{K}, \mathbf{P}_i) - \mathcal{P}_{i-1} \right\| \right), \tag{2}$$

where $\tilde{\mathbf{I}}_i$ and $\tilde{\mathbf{D}}_i$ refer to the inpainted image and its depth map respectively, d_i denotes depth coefficient to be optimized, and M refers to the overlapping region between $\mathcal{P}_i$ and $\mathcal{P}_{i-1}$, as other areas are not shared for calculating ℓ_1 loss.

Thus, we get a set of images refer to the input and indicate specific camera movement:

$$\{\mathbf{I}_0, ..., \mathbf{I}_{N-1}\} = \{\psi(\mathcal{P}_i, \mathbf{K}, \mathbf{P}_i) | i \in [0, N-1]\}. \tag{3}$$

3.2 Layout Prior of Noise

With camera motion modeling, we obtain a sequence $\mathbf{V}_0 = \{\mathbf{I}_0, ..., \mathbf{I}_{N-1}\} \in \mathbb{R}^{N\times 3\times H\times W}$ of rendered images adhering to a specific camera trajectory. Note that quality of rendered images are not perfect since single input image only leads to sparse point cloud reconstruction, besides, these renderings are static, thus they could not use directly as video frames. To form an ideal video, we need to find a way that satisfies three requirements: 1) camera motions should be maintained; 2) video should be encouraged with more dynamics; and 3) quality imperfection should be compensated.

Camera Motion Inversion Recent work on diffusion models have demonstrated the strong controllability of its noisy latents (Meng et al., 2021; Mao et al., 2023), the causality and error-resilience they hold towards final output make them a convenient yet powerful tool for controllable generation of diffusion models. Particularly for initial noise, even sampled from Gaussian distribution, it still have significant influence on the layout of generated image (Mao et al., 2023), so that rearranging

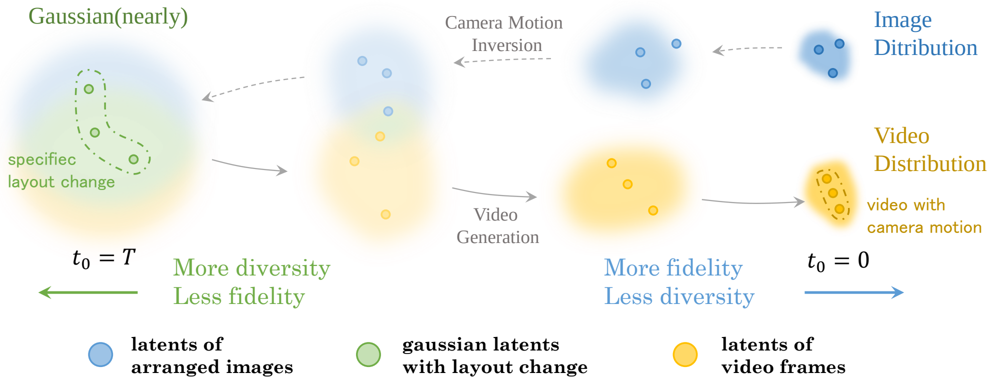


Figure 2: **Transition of samples between two distinct distributions.** As the layout-arranged images being inverted by adding random noise, the distribution of their noisy latents will gradually converge to their video counterpart (green area), finally forming nearly Gaussian latents with specific layout change. This information is then inherited during video generation process. Steps of camera motion inversion determine the trade-off between video diversity and motion fidelity.

the noise pixels will make content in output relocate as well. For instance, if all pixels in initial noise shift to right by a certain distance, it is likely that generated output reflect a similar shift. This reminds us that the impact of camera movement on images could also be regarded as a kind of layout rearrangement, where pixels change their positions caused by viewpoint change. In a similar way, videos can be reorganized following camera motion if their noisy latents change accordingly.

Inspired by this, we first construct a series of noisy latents indicating specific camera movements. It can be intuitively done by employing diffusion's inversion process on rendered image sequence $\mathbf{V}_0$. Latent at timestep t_0 can be calculated as follows, where $\bar{\alpha}_t$ are variances of the scheduler:

$$\mathbf{V}_{t_0} = \sqrt{\bar{\alpha}_{t_0}}\mathbf{V}_0 + \sqrt{1-\bar{\alpha}_{t_0}}\epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{4}$$

Because the rendered images $\mathbf{V}_0$ share common pixels in certain regions, their latents also have relevance to each other in a way indicating pixels' move. Moreover, while being perturbed with random noise, blank spaces and flawed regions in $\mathbf{V}_0$ can be further filled with randomness, providing video model with more possibilities to generate and correct them.

Video Generation After camera motion inversion, noisy latents presenting camera movements are then passed through the backward process of video diffusion model, utilizing their layout controllability to guide video generation. Leveraging prior knowledge of base video model, the generation process also bestows video with rational dynamic information. In this way, explicit camera movements are injected into video diffusion model in an appropriate and training-free fashion. Starting from noisy motion latents at timestep t_0, the generation step can be represented as:

$$\hat{\mathbf{V}}_{t-1} = \sqrt{\alpha_{t-1}}\left(\frac{\mathbf{V}_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(\mathbf{V}_t)}{\sqrt{\alpha_t}}\right) + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\epsilon_\theta^{(t)}(\mathbf{V}_t) + \sigma_t\epsilon, \quad t \in [1, t_0]\,. \tag{5}$$

Here ϵ_θ denotes the video model for noise prediction and σ_t determines whether the denoising process is deterministic or probabilistic, we set $\sigma = 1$ to encourage diversity of generation results.

Trade-off Between Fidelity and Diversity Leveraging noise prior guidance in diffusion model could lead to a trade-off between generation's fidelity and diversity (Meng et al., 2021; Hou et al., 2024), where results that hold more faithfulness towards guidance tend to decline in generation quality. In this task, similar circumstance also occurs as model is required to be guided by some imperfect renderings while generate a reasonable and dynamic video. The key factor to balance the trade-off problem lies in the choice of t_0. When larger t_0 applied, generation bears more resemblance to original guidance $\mathbf{V}_0$, yet lacks of rationality and dynamics to be an appealing video. Instead, smaller t_0 leads to well-generated video, but is less aligned with desired camera motion. Empirically, we find larger t_0 works better for motions with moderate intensity, and for those with relatively drastic move, smaller t_0 shows preferable performance. The process of stage II is illustrated in Fig. 2.

Table 1: **Quantitative comparisons.** Our method attains comparable performance with finetuned methods in both video generation quality and camera motion alignment.

Method	Video Quality				Motion Accuracy		
	FVD ↓	FID ↓	IS ↑	CLIP-SIM ↑	ATE ↓	RPE-T ↓	RPE-R ↓
SVD	1107.93	68.51	7.21	0.3095	4.23	1.79	0.021
MotionCtrl+SVD	<u>810.59</u>	69.03	**7.17**	0.3076	<u>4.19</u>	**1.07**	<u>0.012</u>
CameraCtrl+SVD	951.80	**67.59**	6.82	**0.3138**	4.22	1.17	0.013
CamTrol+SVD	**778.46**	<u>68.06</u>	<u>7.05</u>	<u>0.3110</u>	**4.17**	**1.07**	**0.010**
Reference	-	-	-	-	*3.60*	*0.89*	*0.008*

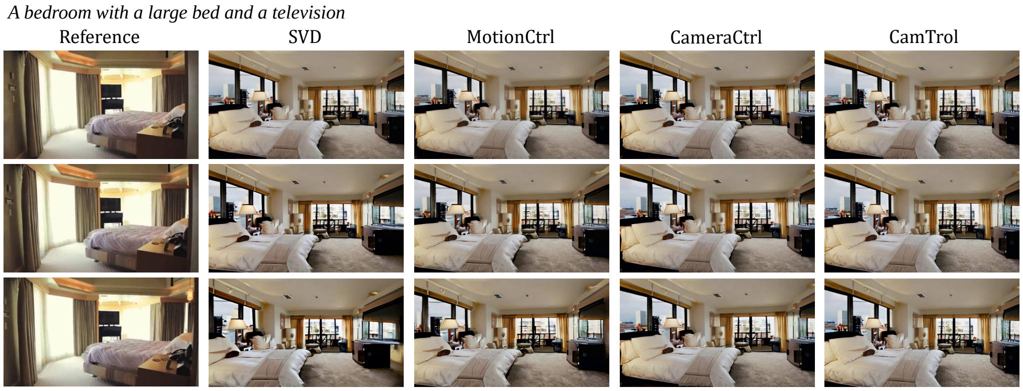


Figure 3: **Qualitative comparisons with finetuned methods.** CamTrol's outputs align well with complex trajectories from reference video, while others fail to perceive subtle camera pose changes. We provide videos in the supplementary materials for a clearer comparison.

4 Experiments

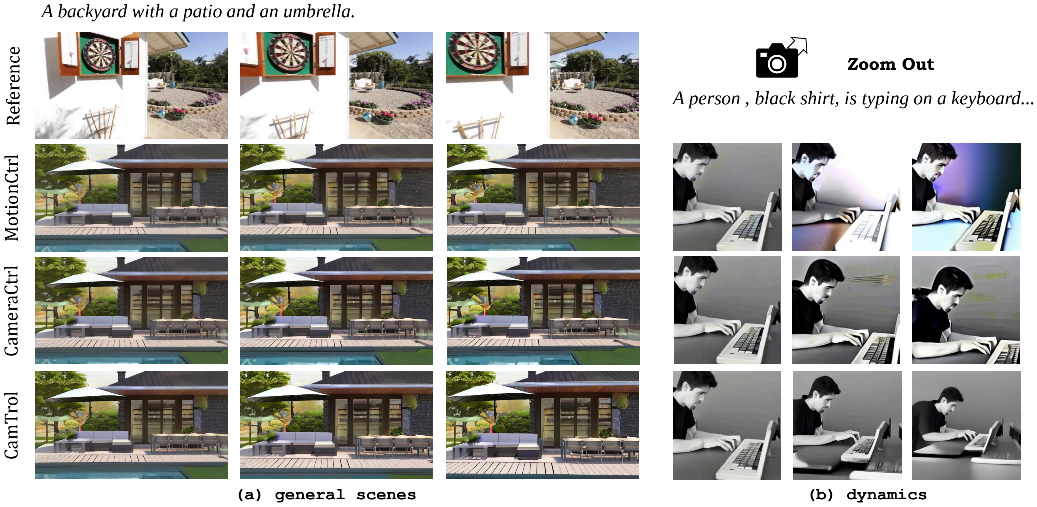
4.1 Experimental Settings

Implementation Details We compare our method with state-of-the-art works including MotionCtrl (Wang et al., 2023) and CameraCtrl (He et al., 2024). To ensure a fair comparison, we employ SVD (Blattmann et al., 2023a) as base model for all methods. SVD is originally trained on resolution of 576 × 1024, but SVD-based CameraCtrl only support 320 × 576. Since changing the original resolution leads to suboptimal generation quality, we use 576 × 1024 for MotionCtrl and CamTrol, then resize their outputs into 320 × 576 for calculating metrics. For all methods, the number of frames and the decoding size of SVD are set to 14. We use 25 steps for both inversion and generation processes.

Evaluation Details In quantitative evaluation, FVD (Unterthiner et al., 2018), FID (Heusel et al., 2017) and IS (Saito et al., 2020) are used to assess video generation quality, and CLIPSIM (Wu et al., 2021) quantifies the similarity between generated video and input prompt. For the accuracy of camera motion, we adopt ParticleSFM(Zhao et al., 2022) and produce estimated camera trajectories from generated videos, with the use of Absolute Trajectory Error(ATE) measuring their differences compared to ground truth. Relative Pose Error (RPE) is calculated to assess between consecutive frames how well the relative motions match expected ones including their transition (RPE-T) and rotation part (RPE-R). Specifically, we randomly sample 500 prompt-trajectory pairs from RealEstate10k (Zhou et al., 2018), and use the corresponding videos as reference for calculating FVD and FID. Since SVD is an image-to-video model, we generate the first frames using Stable Diffusion (Rombach et al., 2022) by text prompts. We also compare the results produces by vanilla SVD as a reference. For camera motion accuracy, we provide the evaluations on ground truth videos as a lower bound of these metrics.

Table 2: **Computational analysis of inference process,** evaluated under unified settings.

		SVD	MotionCtrl	CameraCtrl	CamTrol ($t_0 = 10$)
Max GPU memory(MB)		11542	31702	26208	11542
Time (s)	pre-process	-	-	-	56
	inference	11	32	42	8


A backyard with a patio and an umbrella.
Reference
MotionCtrl
CameraCtrl
CamTrol
Zoom Out
A person , black shirt, is typing on a keyboard...

(a) general scenes

(b) dynamics

Figure 4: **Generalization comparisons.** CamTrol can avoid domain collapse that rises from overfitting on certain datasets, adapting to more general scenes (*Left*, where input does not resemble real scenes), and maintain video's dynamics while adhering to desired camera movements (*Right*).

4.2 Comparisons with State-of-the-art Methods

Quantitative Evaluation Quantitative evaluations are shown in Table 1. In the table, the best results are in bold, and the second best is underlined. The performance of vanilla SVD (without motion control) is indicated as *SVD*, while the lower bound of motion metrics provided by ground truth videos is indicated as *Reference*. In terms of video quality, CamTrol attains comparable performance with methods finetuned on the RealEstate10k dataset. For motion accuracy, CamTrol also gets lowset score in ATE and RPE-T/R. The quantitative evaluation demonstrates CamTrol's ability to generate videos with both accurate camera motion and high visual quality.

Qualitative Analysis Qualitative comparisons are illustrated in Fig. 3. The reference trajectory includes zoom, pan and roll. While MotionCtrl and CameraCtrl fail to perceive subtle camera motions, resulting in simple pan movement, CamTrol is able to follow the complex trajectory and generate video with correct move. We also evaluate the generalization ability of different methods in generating more general scenes and dynamic content, the results are shown in Fig. 4. Since both MotionCtrl and CameraCtrl are finetuned on limited scenes (i.e., real estate videos) with static content, they can hardly generalize to other scenes, such as videos in non-realistic style or videos that include people. As illustrate in Fig. 4, their motion controls in both situations are misaligned with input movements. Furthermore, finetuning on such datasets leads to the loss of dynamics, which is a crucial element in video generation. In comparison, CamTrol preserves most of the prior knowledge from video base models, enabling it to handle general scenes as well as generate dynamic content. Relevant videos are available in the supplementary materials.

Computational Analysis We provide the computational analysis in Table 2, including the maximum GPU memory required and the time consumption for all methods during inference process. The evaluations are conducted under unified settings. We test on 576×320, with 25 generation steps, the number of frames and decoding size of SVD are set to 14. As a training-free method, CamTrol

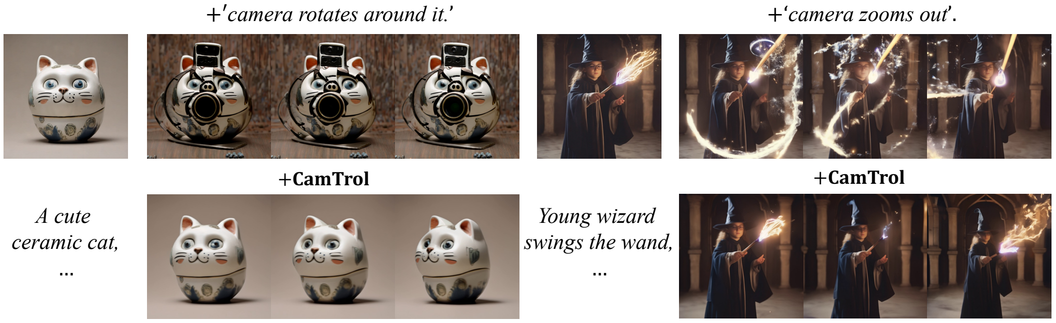

+'camera rotates around it.'
+'camera zooms out'.
+CamTrol
+CamTrol
A cute ceramic cat, ...
Young wizard swings the wand, ...

Figure 5: **Comparison with base model.** Controlling camera motion via prompt engineering doesn't work at most times. Instead, CamTrol offers robust control towards video's camera movement in a training-free manner.

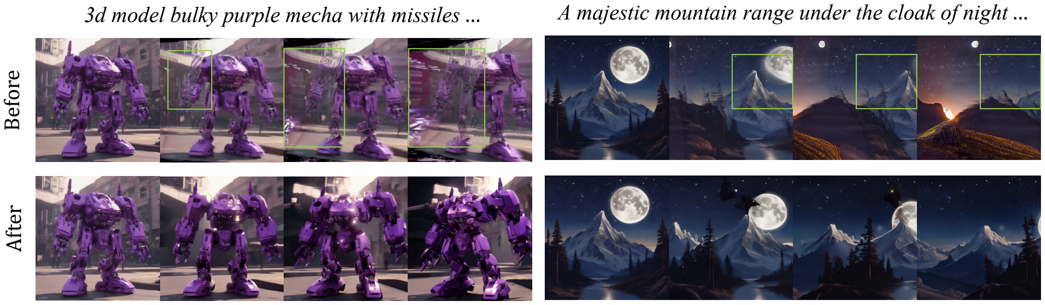

3d model bulky purple mecha with missiles ...
A majestic mountain range under the cloak of night ...
Before
After

Figure 6: **Effectiveness of layout prior.** Layout prior guidance compensates for the vacancies (*Left*) and static contents (*Right*) caused by point cloud rendering.

requires no extra GPU memory during inference process compared with base model. This makes it save 10-20GB of GPU memory than other methods under the same circumstances, and makes it applicable running on a single RTX 3090. The major consumption of CamTrol comes from the time rendering multi-view images. Since this process is currently sequential in time, i.e., $t_0 \rightarrow t_1 \rightarrow t_2$, a more parallel approach may be more time-efficient. We will leave this to future works. The results of 576×1024 resolution and more detailed settings can be found in the Appendix.

4.3 Ablation Study

Comparison to Base Model To demonstrate that changes of camera motion are attributed to our method rather than the innate capability of video model, we conduct ablation study to assess its effectiveness. We add prompts describing certain camera moves (e.g. *zooms out*), letting video model understand by itself. The results are shown in Fig. 5. It could be observed that even provided with prompts indicating how camera should move, base model fail to produce correct results. Instead, CamTrol is able to implement designated motion control without any instructions from text prompts. In Table 1, the comparison with vanilla SVD also demonstrate CamTrol's effectiveness.

Effectiveness of Layout Prior We employ ablation study to validate the effectiveness of layout prior guidance, illustrating its necessity from two aspects: the completeness of vacancies and dynamic of generated video. In Fig. 6, we showcase frames before and after noise prior guidance. With camera pose changes, there appears regions unfilled in point cloud and causes blank spaces in rendered images(left part); Besides, due to the static nature of point cloud, rendered images remain stationary(right part). Noise layout prior could compensate for these flaws, finally produce videos with inpainted vacancies and rationalized dynamics.

Table 3: **Quantitative effect of t_0.**

t_0	Video Quality				Motion Accuracy		
	FVD ↓	FID ↓	IS ↑	CLIP-SIM ↑	ATE ↓	RPE-T ↓	RPE-R ↓
$t_0 = 5$	1079.88	68.52	7.14	0.3100	4.17	1.09	0.012
$t_0 = 10$	778.46	68.06	7.05	0.3110	4.17	1.07	0.010
$t_0 = 15$	754.14	67.98	7.00	0.3107	4.13	1.02	0.008

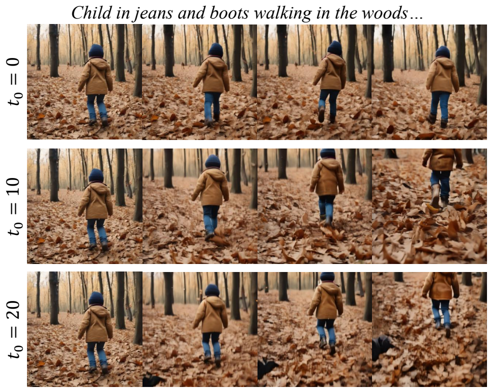


Figure 7: **Effect of t_0.** Smaller t_0 encourages dynamics while larger t_0 preserves camera movements (*Pedestal Down*).

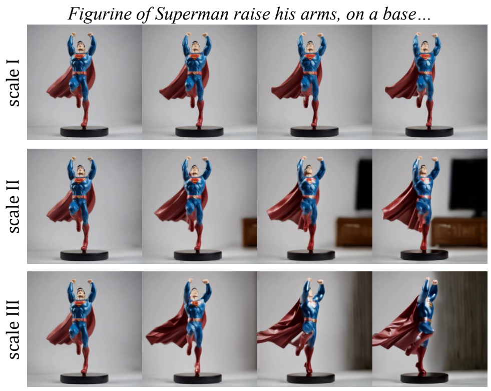


Figure 8: **3D rotation videos at different motion scales.** CamTrol supports camera movements over various scales.

Effect of Timestep t_0 t_0 is a crucial factor that influences the trade-off between generated video's diversity and its faithfulness to camera motions requirements. To investigate its effect on the output, we conduct experiments with various t_0 values, relevant results are shown in Fig. 7. As illustrated, videos generated with larger t_0 tend to conform better to camera motion requirements, but suffer from decrease in dynamics; On the contrary, smaller t_0 leads to more plausible generations but fails to meet camera's requirements, as latents at these timesteps carry more randomness. We also provide quantitative evaluations of different t_0 in Table 3. As t_0 increase, CamTrol produces videos alike to static, camera-moving scenes (which have lower FVD as we use RealEstate10k as reference videos) with higher accuracy in motion control.

Generalization to Diverse Situations Our proposed CamTrol can be seamlessly plugged and played under various scenarios, accommodating different base models, resolutions, generating length, all in training-free style. We present visual results of its applications under different settings, including CogVideoX (Yang et al., 2024b)(diffusion transformer model, 720 × 480 resolution, 49 frames) and VideoFusion (Luo et al., 2023)(decomposed diffusion process, 128 × 128 resolution, 16 frames), in Fig. 10. Our approach remains effective applied to alternative video base models, resolutions and generating lengths, demonstrating its strong robustness and generalization ability.

4.4 Further Applications

3D Rotation Videos One of the most advantages of our method is it can generate videos with rotating moves and produce outputs similar to 3D generation models (Voleti et al., 2024; Melas-Kyriazi et al., 2024). While these 3D models need large-scale training on 3D dataset and could only handle inputs in specific styles, our approach is able to deal with any type of images and achieve this in a completely zero-shot manner. One example of this is shown in Fig. 8.

Hybrid and Complex Camera Movements By combining different basic camera trajectories, CamTrol can support hybrid camera movements and endow generated video with cinematic charm. Besides this, explicit motion modeling also equips CamTrol with the ability to support trajectories containing precise extrinsics, and generate videos presenting any complicated camera movement.

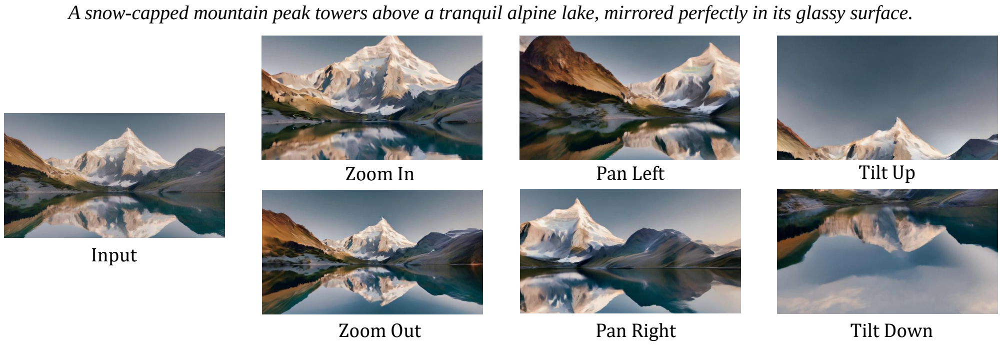
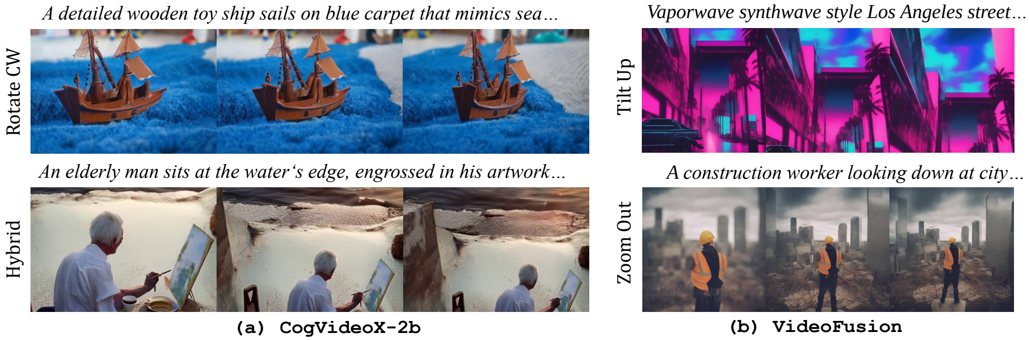

A snow-capped mountain peak towers above a tranquil alpine lake, mirrored perfectly in its glassy surface.
Input
Zoom In
Pan Left
Tilt Up
Zoom Out
Pan Right
Tilt Down

Figure 9: **Multi-trajectory video generation.** CamTrol is able to generate videos with different trajectories for one scene.


A detailed wooden toy ship sails on blue carpet that mimics sea…
Rotate CW
An elderly man sits at the water's edge, engrossed in his artwork…
Hybrid
Vaporwave synthwave style Los Angeles street…
Tilt Up
A construction worker looking down at city…
Zoom Out

(a) CogVideoX-2b

(b) VideoFusion

Figure 10: **Applied onto CogVideoX (Yang et al., 2024b) and VideoFusion (Luo et al., 2023).** CamTrol can be plugged and played under various situations, accommodating different base models, different resolutions, different generating lengths, all in a training-free manner.

Multi-trajectory Video Generation One natural application of our method is to generate multi-trajectory videos for one scene. Since the point cloud is fixed once the reconstruction process finished, the consistency of the content between different trajectories is guaranteed, and we can easily produce multiple camera-moving videos of this scene. We present this application in Fig. 9. More results on multi-trajectory video generation could be found in the supplementary materials.

Camera Motion at Different Scales CamTrol supports camera movements at controllable scales. By specifying different magnitudes of camera's extrinsic matrix within point cloud spaces, rendered images will exhibit varying degrees of motion, leading to videos with distinct scales of camera movements. This further demonstrate the powerful controllablity of CamTrol, and provides a new pathway for video's customized camera control. We provide relevant visualization in Fig. 8.

5 Conclusion

In this paper, we propose a training-free and robust method **CamTrol** to offer camera control for off-the-shelf video diffusion models. It consists of two-stage procedure including explicit camera motion modeling in 3D point cloud space and video generation utilizing layout prior of noisy latents. Compared to previous work, CamTrol does not require any additional finetuning on camera-annotated datasets or self-supervised training via data augmentation, simultaneously, comprehensive experiments demonstrate its superior performance in both generation video quality and motion alignment against other state-of-the-art methods. Furthermore, we show the ability of CamTrol generalizing to various scenarios, as well as its impressive applications including unsupervised generation of 3D rotation video, scalable motion control and dealing with complicated camera trajectories.

REFERENCES

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.

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*, 2023a.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22563–22575, 2023b.

T Brooks, B Peebles, C Homes, W DePue, Y Guo, L Jing, D Schnurr, J Taylor, T Luhman, E Luhman, et al. Video generation models as world simulators, 2024.

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.

Zilong Chen, Yikai Wang, Feng Wang, Zhengyi Wang, and Huaping Liu. V3d: Video diffusion models are effective 3d generators. *arXiv preprint arXiv:2403.06738*, 2024.

Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14367–14376, 2021.

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.

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

Hao Fei, Shengqiong Wu, Wei Ji, Hanwang Zhang, and Tat-Seng Chua. Empowering dynamics-aware text-to-video diffusion with large language models. *arXiv preprint arXiv:2308.13812*, 2023.

Ruoyu Feng, Wenming Weng, Yanhui Wang, Yuhui Yuan, Jianmin Bao, Chong Luo, Zhibo Chen, and Baining Guo. Ccedit: Creative and controllable video editing via diffusion models. *arXiv preprint arXiv:2309.16496*, 2023.

Songwei Ge, Seungjun Nah, Guilin Liu, Tyler Poon, Andrew Tao, Bryan Catanzaro, David Jacobs, Jia-Bin Huang, Ming-Yu Liu, and Yogesh Balaji. Preserve your own correlation: A noise prior for video diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22930–22941, 2023.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *The Twelfth International Conference on Learning Representations*, 2023.

Junlin Han, Filippos Kokkinos, and Philip Torr. Vfusion3d: Learning scalable 3d generative models from video diffusion models. *arXiv preprint arXiv:2403.12034*, 2024.

Zekun Hao, Xun Huang, and Serge Belongie. Controllable video generation with sparse trajectories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7854–7863, 2018.

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.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022.

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.

Chen Hou, Guoqiang Wei, and Zhibo Chen. High-fidelity diffusion-based image editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 2184–2192, 2024.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2426–2435, 2022.

Xin Li, Wenqing Chu, Ye Wu, Weihang Yuan, Fanglong Liu, Qi Zhang, Fu Li, Haocheng Feng, Errui Ding, and Jingdong Wang. Videogen: A reference-guided latent diffusion approach for high definition text-to-video generation. *arXiv preprint arXiv:2309.00398*, 2023.

Shaoteng Liu, Yuechen Zhang, Wenbo Li, Zhe Lin, and Jiaya Jia. Video-p2p: Video editing with cross-attention control. *arXiv preprint arXiv:2303.04761*, 2023.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461–11471, 2022.

Zhengxiong Luo, Dayou Chen, Yingya Zhang, Yan Huang, Liang Wang, Yujun Shen, Deli Zhao, Jingren Zhou, and Tieniu Tan. Videofusion: Decomposed diffusion models for high-quality video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10209–10218, 2023.

Ze Ma, Daquan Zhou, Chun-Hsiao Yeh, Xue-She Wang, Xiuyu Li, Huanrui Yang, Zhen Dong, Kurt Keutzer, and Jiashi Feng. Magic-me: Identity-specific video customized diffusion. *arXiv preprint arXiv:2402.09368*, 2024.

Jiafeng Mao, Xueting Wang, and Kiyoharu Aizawa. Guided image synthesis via initial image editing in diffusion model. In *Proceedings of the 31st ACM International Conference on Multimedia*, pp. 5321–5329, 2023.

Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, Natalia Neverova, Andrea Vedaldi, Oran Gafni, and Filippos Kokkinos. Im-3d: Iterative multiview diffusion and reconstruction for high-quality 3d generation. *arXiv preprint arXiv:2402.08682*, 2024.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pp. 16784–16804. PMLR, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Masaki Saito, Shunta Saito, Masanori Koyama, and Sosuke Kobayashi. Train sparsely, generate densely: Memory-efficient unsupervised training of high-resolution temporal gan. *International Journal of Computer Vision*, 128(10-11):2586–2606, 2020.

Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023.

Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022.

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.

Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. *arXiv preprint arXiv:2403.12008*, 2024.

Zhouxia Wang, Ziyang Yuan, Xintao Wang, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. *arXiv preprint arXiv:2312.03641*, 2023.

Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. Godiva: Generating open-domain videos from natural descriptions. *arXiv preprint arXiv:2104.14806*, 2021.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7623–7633, October 2023.

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. *arXiv preprint arXiv:2402.03162*, 2024a.

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*, 2024b.

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.

Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9150–9161, 2023.

Yan Zeng, Guoqiang Wei, Jiani Zheng, Jiaxin Zou, Yang Wei, Yuchen Zhang, and Hang Li. Make pixels dance: High-dynamic video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Wang Zhao, Shaohui Liu, Hengkai Guo, Wenping Wang, and Yong-Jin Liu. Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild. In *European Conference on Computer Vision*, pp. 523–542. Springer, 2022.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817*, 2018.

A More Results on Camera Control

We showcase additional qualitative results of CamTrol on various camera movements. **As pdf format is not the best way to exhibit videos, we put them on our anonymous demo page:**

Please Click Here for Our Video Results.

If not accessible, please refer to the supplementary materials for a quicker view to the demo page.

This demo page includes CamTrol-generated videos including basic camera motions (including *Zoom, Tilt, Pan, Pedestal, Truck, Roll, Rotate, Hybrid, Complicated*, detailed definitions are in B), hybrid motions (*Zoom In first, then Pedestal Up*, *Zoom Out* + *Pedestal Up* + *Truck Left* + *Tilt Down* + *Pan Right*) and complicated motions generated from precise camera extrinsics (extracted from RealEstate10k (Zhou et al., 2018)).

Besides, it contains 3D rotation videos generated unsupervisedly from video base models (both objects and scenes). These outputs share sort of similarity with outputs of 3D generation models, as they all exhibit in a turning-table like way, which camera rotates around some objects. The difference here is that 3D model, as trained on specific datasets, could only generate outputs in certain styles, *e.g.*, single static object with no background. Instead, our model could handle arbitrary image as input, and generate a rotating-around video with proper dynamics. From this aspect, our method could be seen as a infinite source of attaining 3D data. And by utilizing our method with stronger backbones, video foundation models could truly become the largest source of 3D data as it should be.

Furthermore, it contains additional results mentioned in ablation studies, e.g. controlling camera motions at different scales, and the effectiveness of using layout prior compared with the raw output given by video base model.

B Definitions of Basic Camera Motions

We refer to the terminology in cinematography to describe different camera motions, definitions of each type are detailed in Table 5.

To get consistent images from multiple views, we set camera motion as a trajectory of extrinsic matrix $\{\mathbf{P}_1, ..., \mathbf{P}_{N-1}\}$, each including a rotation matrix and translation matrix representing camera's pose and position. For hybrid motions, CamTrol supports both spatial (i.e. moving several basic moves simultaneously) and temporal (concatenation of basic motions sequentially) combinations. In the case of complicated trajectories, one can directly use precise parameters for camera's extrinsic matrix as input to control video's motion. Additionally, if these parameters not available, a reference video with corresponding move can also serve as input. With the help of SFM algorithms (e.g. COLMAP [1], ParticleSFM (Zhao et al., 2022)), camera motion can be easily estimated and used for imitation. In this sense, CamTrol can also be seen as an unsupervised method to achieve video motion transfer.

C Visual Comparisons with State-of-the-arts

Sec. 3 showcases some qualitative comparisons between CamTrol and different state-of-the-art approaches. For better visualization and comparison, we provide more video results in the supplementary materials. Please check them if needed.

D Implementation Details

For text prompt input, we use Stable Diffusion v2-1 [2] or Stable Diffusion XL [3] to generate the initial image. The inpainting model we apply is Stable Diffusion inpainting model proposed by runway

[1] https://github.com/colmap/colmap

[2] https://huggingface.co/stabilityai/stable-diffusion-2-1

[3] https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0

Table 4: **Computational analysis of on 576 × 1024.**

		SVD	MotionCtrl	CameraCtrl	CamTrol ($t_0 = 10$)
Max GPU memory(MB)		34236	71096	-	34236
Time (s)	pre-process	-	-	-	149
	inference	32	54	-	22

Table 5: **Definitions of camera motions.** We follow the terminology in cinematography to describe different camera movements.

Camera Motion	**Directions**	**Definition**
Zoom	In Out	Camera moves towards or away from a subject.
Tilt	Up Down	Rotating the camera vertically from a fixed position.
Pan	Left Right	Rotating the camera horizontally from a fixed position.
Pedestal	Up Down	Moving a camera vertically in its entirety.
Truck	Left Right	Moving a camera horizontally in its entirety.
Roll	Clockwise Anticlockwise	Rotating a camera in its entirety in a horizontal manner.
Rotate	Clockwise Anticlockwise	Moving a camera around a subject.
Hybrid	Combinations	Spatial and temporal combination of basic motions.
Complicated	Arbitrary	Camera extrinsic matrix or a reference video as input.

[4], and backward step of inpainting is set to 25. We deploy ZeoDepth [5] as depth estimation model. The classifier-free guidance scale follows default setting of base models themselves. We use SVD's default setting of 6 fps for video generation, and process reference videos to 6 fps for FVD and FID calculation. The complete procedure of CamTrol is elucidated in Algorithm 1.

For computational analysis, we set both number of frames and decoding size to 14, generation steps to 25. We do not apply xformers in all approaches. The computational analysis at 576 × 1024 resolution is shown in Table 4. SVD-based CameraCtrl only support resolution at 320 × 576.

E Choice of 3D Representation

In Sec. 3.1, we designate point cloud as the intermediate 3D representation for explicit camera motion modeling. Doubts may arise why we do not use more complex 3D representation which might be more useful. Here we take the most prevalent 3D representation: 3D Gaussian Splatting [6], as example to elaborate the benefit of using point cloud in three aspects:

Firstly, concerning the input, point cloud is able to construct the whole scene from one single input image combining techniques of depth estimation and inpainting. GS, though also an explicit 3D representation, requires optimization following images from different views, which means it is

[4] https://huggingface.co/runwayml/stable-diffusion-inpainting

[5] https://github.com/isl-org/ZoeDepth

[6] https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

Algorithm 1: Training-free camera control for video generation

```
Input: Text prompt $p$, camera motion $\mathbf{P}$, input image $\mathbf{I}_0$ (optional).
// Stage I: Camera Motion Modeling:
for i=1,...,N-1 do
    $\tilde{\mathbf{I}}_i = \text{inpainting}(\mathbf{I}_{i-1}, \mathbf{P}_i, p)$ ;
    $\tilde{\mathbf{D}}_i = \text{depth}(\tilde{\mathbf{I}}_i)$ ;
    while not converged do
        $d_i = \text{argmin}_d \left( \sum_M \left\| \phi([\tilde{\mathbf{I}}_i, d\tilde{\mathbf{D}}_i], \mathbf{K}, \mathbf{P}_i) - \mathcal{P}_{i-1} \right\| \right)$
    end
    $\mathcal{P}_i = \phi([\tilde{\mathbf{I}}_i, d_i\tilde{\mathbf{D}}_i], \mathbf{K}, \mathbf{P}_i)$ ;
    $\mathbf{I}_i = \psi(\mathcal{P}_i, \mathbf{K}, \mathbf{P}_i)$ ;
end
// Stage II: Layout Prior Generation:
$\mathbf{V}_0 \leftarrow \{\mathbf{I}_i\}_{i=0}^{N-1}$ ;
Sample random noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ;
Motion inversion $\mathbf{V}_{t_0} \leftarrow \sqrt{\bar{\alpha}_{t_0}}\mathbf{V}_0 + \sqrt{1 - \bar{\alpha}_{t_0}}\epsilon$ ;
for $t$=$t_0$,...,1 do
    $\mathbf{V}_{t-1} \leftarrow \sqrt{\alpha_{t-1}} \left( \frac{\mathbf{V}_t - \sqrt{1-\alpha_t}\epsilon_\theta^{(t)}(\mathbf{V}_t)}{\sqrt{\alpha_t}} \right) + \sqrt{1 - \alpha_{t-1} - \sigma_t^2}\epsilon_\theta^{(t)}(\mathbf{V}_t) + \sigma_t\epsilon$
end
```

neither capable of handling single input image, nor can it leverages 2D inpainting to complete the scene gradually.

Secondly, from the aspect of time, point cloud can directly lift 2D points onto 3D spaces, while 3DGS demands optimization on each scenario. As a training-free method, our method takes nearly no time to generate a video after multi-view images are acquired, but would need more time if 3DGS were applied.

Lastly, from the task itself, the target of stage I is not precisely reconstruct the 3D scene but only offer a rough layout guidance, in this context, rendered images from point cloud qualify enough and no further refinement on 3D reconstruction is necessary.

From the analysis, point cloud is quite qualified serving as a rough layout guidance in a relatively quick speed, without any further optimization and redundant multi-view images as input. The use of point cloud allows our algorithm to be totally training-free and optimization-free, simultaneously being able to produce camera-moving videos quickly with merely one single image or text prompt as input.

F Details about Generating 3D Rotation Videos

3D generation models (Voleti et al., 2024; Melas-Kyriazi et al., 2024; Shi et al., 2023; Chen et al., 2024; Han et al., 2024) are trained on highly-regulated 3D datasets, these datasets are hard to collect, and consist only a limited variety of data types (e.g. single static objects with no background). As a consequence, 3D generation models exhibit very constrained output distribution and could only produce results in certain styles. CamTrol avoid these problems of arduous data collecting, laborious finetuning and output collapse by utilizing the layout prior hidden in video foundation models. Not only does it require no training, but this advantage also benefits CamTrol from inheriting most of the prior knowledge inside video foundation models, e.g., the diversity of scenarios, the dynamic of moving objects, temporal consistency, etc. Thus CamTrol is able to generate dynamic 3D content, both objects and scenes, in a totally unsupervised and training-free manner, this is what other method cannot achieve yet.

Compared with regulated datasets, the problem of processing wild pictures in 3D is that some of the parameters is unknown. In Sec. 3.1, we've mentioned that the camera intrinsic matrix $\mathbf{K}$ and initial extrinsic matrix $\mathbf{P}_0$ are set by convention as they're usually intractable. Another crucial parameter concerning 3D video generation is the distance between camera and the content of input image (denote as f), note that input image could be synthetic or real. Considering most camera rotations are done around the center point, we extract a patch from the very center of the input image, conducting depth estimations on the it and define the distance f as the averaged depth. The rotation and transition matrix during camera moving can be formed as :

$$\begin{gathered}\mathcal{R}_y = \begin{bmatrix} \cos\theta_i & 0 & -\sin\theta_i \\ 0 & 1 & 0 \\ \sin\theta_i & 0 & \cos\theta_i \end{bmatrix}, \quad t = \begin{bmatrix} f\sin\theta_i \\ 0 \\ f - f\cos\theta_i \end{bmatrix}, \\ \text{where} \quad f = \frac{1}{|P|}\sum_{(j,k)\in P} D(x_0 + j, y_0 + k). \end{gathered} \tag{6}$$

Here $i \in [0, N-1]$ and θ_i refers to rotation angle around y axis at step i, D denotes depth estimation of image, P represents the patch around central point (x_0, y_0). In our experiment, we choose $(j, k) \in [-10, 10]$ as the size of patch.

G Motion Blur, Problems and Solutions

Videos produced by CamTrol need to satisfy certain camera movement, and some drastic perspective changes might cause visible trails, recognised as motion blur of objects or scenes. This phenomenon will appear to be more indispensable when video base model holds a relatively small generation length (e.g. 16 frames) as well as the motion scale becomes larger (e.g. tilt up for 60 degrees or more). To avoid blur issues in controlling video camera motion, we propose several solutions as below:

1. According to the analysis above, the blur issue is caused by limited generation frames and large camera movement, thus the most intuitive solution is to either cut down motion scale or utilize a more capable generation model. For severe perspective changes, the optimal approach would involve employ video foundation models that support larger generation length (e.g. CogVideoX (Yang et al., 2024b) supports generating video with 49 frames). This allows model to manage motions of equivalent magnitude while experiencing a smaller moving range between adjacent frames, thereby brings effective alleviation to blur problems.
2. One can also stack the results of multiple generations to form a complete outcome, i.e., treating the last frame of previous generation as the starting frame for the next, and integrate them as a whole. This approach is more suitable when utilizing a image-to-video (I2V) base model. Since most open-source video foundation models are text-to-video (T2V), one may consider increasing the step of camera motion inversion t_0, which guarantees more fidelity towards input images' content (and motion).
3. Besides the above two approaches, it is as well a common and convenient choice to apply frame interpolation towards the output. Lots of off-the-shelf frame interpolation models are open-sourced and could be found on github.
4. Lastly, if video length is not being able to alter, it may be necessary to increase the fps of generated videos for better visualization. Although larger fps leads to shorter video duration, it simultaneously makes the visual persistence brought by motions less pronounced, which reduce blur visually.

In our experiments, we take raw output in all settings.