VD3D: TAMING LARGE VIDEO DIFFUSION TRANSFORMERS FOR 3D CAMERA CONTROL

Anonymous authors
Paper under double-blind review

Figure 1: **3D camera control for text-to-video generation.** We introduce a method that can control camera poses for text-to-video generation using video diffusion transformers. (left) The method takes as input a set of camera poses used to generate each frame of a rendered video. (center, right) Applying multiple camera trajectories with the same text prompt enables synthesis of complex scenes from a varied set of viewpoints.

ABSTRACT

Modern text-to-video synthesis models demonstrate coherent, photorealistic generation of complex videos from a text description. However, most existing models lack fine-grained control over camera movement, which is critical for downstream applications related to content creation, visual effects, and 3D vision. Recently, new methods demonstrate the ability to generate videos with controllable camera poses—these techniques leverage pre-trained U-Net-based diffusion models that explicitly disentangle spatial and temporal generation. Still, no existing approach enables camera control for new, transformer-based video diffusion models that process spatial and temporal information jointly. Here, we propose to tame video transformers for 3D camera control using a ControlNet-like conditioning mechanism that incorporates spatiotemporal camera embeddings based on Plücker coordinates. The approach demonstrates state-of-the-art performance for controllable video generation after fine-tuning on the RealEstate10K dataset. To the best of our knowledge, our work is the first to enable camera control for transformer-based video diffusion models.

1 INTRODUCTION

Text-to-video foundation models achieve unprecedented visual quality (Brooks et al., 2024; Sharma et al., 2024). They are trained on massive collections of images and videos and learn to synthesize remarkably consistent and physically plausible visualizations of the world. Yet, they lack built-in mechanisms for explicit 3D control during the synthesis process, requiring users to manipulate outputs through prompt engineering and trial and error—a slow, laborious, and computationally expensive process. For example, as Fig. 2 shows, state-of-the-art video models struggle to follow even simple "zoom-in" or "zoom-out" camera trajectories using text prompt instructions (see supplemental webpage). This lack of controllability limits interactivity and makes existing video generation techniques challenging to use for artists or other end users. We augment 2D video generation models

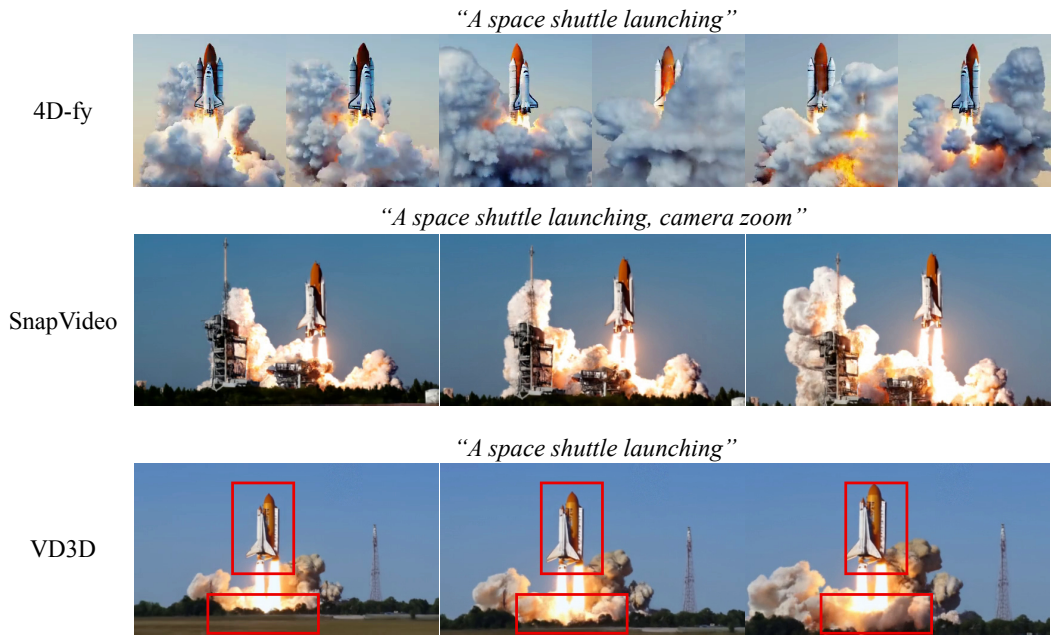

Figure 2: **Comparing text-to-video, text-to-4D, and camera-conditioned text-to-video generation.** We show time progressing across columns for generated videos from different approaches. Text-to-4D approaches, such as 4D-fy (Bahmani et al., 2024b) (top), have complete control over the camera through a 3D representation but lack photorealism. We visualize time progressing while simultaneously changing the viewpoint. (middle) Methods for text-to-video generation (Menapace et al., 2024) create realistic videos but do not provide explicit control over the viewpoint. In contrast, camera-conditioned text-to-video generation (bottom) bridges the gap between the two paradigms by extending text-to-video generators with 3D camera control without using an explicit 3D representation. Specifically, we condition VD3D with a zoom-in camera trajectory to control the generation.

with control over the position and orientation of the camera, providing finer-grained control compared to text prompting, and facilitating use of video generation models for downstream applications.

Several contemporary works (Wang et al., 2023e; He et al., 2024a; Yang et al., 2024b) propose methods for camera control of state-of-the-art, open-source video diffusion models. The key technical insight proposed by these methods is to add camera control by fine-tuning the temporal conditioning layers of a U-Net-based video generation model on a dataset with high-quality camera annotations. While these techniques achieve promising results, they are not applicable to more recent, high-quality transformer-based architectures (Vaswani et al., 2017; Peebles & Xie, 2023), such as Sora (Brooks et al., 2024), SnapVideo (Menapace et al., 2024), and Lumina-T2X (Gao et al., 2024a), as these latest works simply do not have standalone temporal layers amenable to camera conditioning.

Large video transformers represent a video as a (possibly compressed) sequence of tokens, applying self-attention layers to all the tokens jointly (Brooks et al., 2024; Menapace et al., 2024). Consequently, as diffusion transformers do not have standalone temporal layers, they are incompatible with current camera conditioning approaches. As the community shifts towards large video transformers to jointly model spatiotemporal dependencies in the data, it is critical to develop methods that provide similar capabilities for camera control. Our work designs a camera conditioning method tailored to the joint spatiotemporal computation used in large video transformers and takes a step towards taming them for controllable video synthesis.

We develop our work on top of our implementation of SnapVideo (Menapace et al., 2024), a state-of-the-art video diffusion model, that employs Far-reaching Interleaved Transformers (FIT) blocks (Chen & Li, 2023) for efficient video modeling in the compressed latent space. We investigate various camera conditioning mechanisms in the fine-tuning scenario and explore trade-offs in terms of visual quality preservation and controllability. Our findings reveal that simply adapting existing approaches to video transformers does not yield satisfactory results: they either enable some limited amount

of control while reducing the visual quality of the output video, or they entirely fail to control camera motion. Our key technical insight is to enable the control through spatiotemporal camera embeddings, which we derive by combining Plücker coordinates with the network input through a separately trained cross-attention layer. To the best of our knowledge, our work is the first to explore a ControlNet-like (Zhang et al., 2023) conditioning mechanism for spatiotemporal transformers.

We evaluate the method on a collection of manually crafted text prompts and unseen camera trajectories and compare to baseline approaches that incorporate previous camera control methods into a video transformer. Our approach achieves state-of-the-art results in terms of camera controllability and video quality, and also enables downstream applications such as multi-view, text-to-video generation, as depicted in Figure 1. In contrast to existing image-to-3D methods (e.g., (Liu et al., 2024a; Qian et al., 2024b; Voleti et al., 2024a)), which are limited to object-centric scenes, our approach synthesizes novel views for real input images with complex environments.

Overall, our work makes the following contributions.

- We propose a new method to tame large video transformers for 3D camera control. Our approach uses a ControlNet-like conditioning mechanism that incorporates spatiotemporal camera embeddings based on Plücker coordinates.
- We thoroughly evaluate this approach, including comparisons to previous camera control methods, which we adapt to the video transformer architecture.
- We show state-of-the-art results in camera-controllable video synthesis by applying the proposed conditioning method and fine-tuning scheme to the SnapVideo-based model (Menapace et al., 2024).

2 Related work

Our method is connected to techniques related to text-to-video, text-to-3D, and text-to-4D generation. As this is a popular and fast-moving field, this section provides only a partial overview with a focus on the most relevant techniques; we refer readers to Po et al. (2023) and Yunus et al. (2024) for a more thorough review of related techniques.

Text-to-video generation. Our work builds on recent developments in 2D video generation models. One such class of these techniques works by augmenting text-to-image models with layers that operate on the temporal dimension to facilitate video generation (Blattmann et al., 2023b; Singer et al., 2023a; Wu et al., 2023; Guo et al., 2024; Blattmann et al., 2023a). Video generation models can be trained in a hybrid fashion on both images and videos to improve the generation quality (Bain et al., 2021; Wang et al., 2023b; Xue et al., 2022; Ho et al., 2022a; Guo et al., 2024; He et al., 2022; Wang et al., 2023c; Zhou et al., 2022). While they are primarily based on convolutional, U-Net-style architectures, a recent shift towards transformer-based architectures enables synthesis of much longer videos with significantly higher quality (Brooks et al., 2024; Ma et al., 2024b; Menapace et al., 2024; Ma et al., 2024a). Still, these methods do not enable synthesis with controllable camera motion.

4D generation. Previous methods also tackle the problem of 4D generation, i.e., generating videos of dynamic 3D scenes from controllable viewpoints, usually from an input text prompt or image. Since the initial work on this topic using large-scale generative models (Singer et al., 2023b), significant improvements in the visual quality and motion quality of generated scenes have been achieved (Ren et al., 2023; Ling et al., 2024a; Bahmani et al., 2024b; Zheng et al., 2024; Bahmani et al., 2024a). While these methods generate scenes based on text conditioning, other approaches convert an input image or video to a dynamic 3D scene (Ren et al., 2023; Zhao et al., 2023; Yin et al., 2023; Pan et al., 2024; Zheng et al., 2024; Ling et al., 2024a; Gao et al., 2024b; Zeng et al., 2024; Chu et al., 2024). Another line of work (Bahmani et al., 2023a; Xu et al., 2023) extends 3D GANs into 4D by training on 2D videos, however the quality is limited and models are trained on single category datasets. All of these methods are focused on object-centric generation, typically based on 3D volumetric representations. As such, they typically do not incorporate background elements, and overall, they do not approach the level of photorealism demonstrated by the state-of-the-art video generation models used in our technique (see Fig. 2).

Controllable generation with diffusion models. Methods for controllable generation using diffusion models have had significant impact, both in the context of image and video generation. For example,

existing techniques allow controllable image generation conditioned on text, depth maps, edges, pose, or other signals (Zhang et al., 2023; Ye et al., 2023). While Chen et al. (2024) developed a ControlNet-based mechanism for transformer-based diffusion, limited to spatial conditioning with spatial signals, we explore conditioning mechanisms for camera poses (a spatio-temporal signal with an intricate temporal component) in a spatio-temporal transformer. Furthermore, there is a line of work for 3D generation that conditions diffusion models on camera poses for view-consistent multi-view generation (Watson et al., 2023; Tseng et al., 2023; Chan et al., 2023; Yu et al., 2023a; Kumari et al., 2024; Müller et al., 2024; Gao et al., 2024c). Our approach is most similar to related techniques in video generation that seek to control the camera position. For example, MotionCtrl (Wang et al., 2023e) designs camera and object control mechanisms for the VideoCrafter1 (Chen et al., 2023b) and SVD (Blattmann et al., 2023a) models. However, MotionCtrl is designed for U-Net-based approaches and does not directly apply to video diffusion transformers.

Concurrent 3D camera control methods. Concurrent approaches enable camera control by conditioning the temporal layers of the network with camera pose information, e.g., using Plücker coordinates (He et al., 2024a; Guo et al., 2024; Xu et al., 2024b; Kuang et al., 2024) or other embeddings (Yang et al., 2024b). Interestingly, it is also possible to incorporate camera control into video generation models without additional training through manipulation and masking of attention layers, though this requires additional tracking, segmentation, or depth for each input video (Hu et al., 2024; Xiao et al., 2024; Hou et al., 2024). Another recent work (Ling et al., 2024b) transfers motion, including camera motion, to other generated videos.

Although these approaches demonstrate promising results for U-Net-based video diffusion models, the techniques are not applicable to modern video transformers that model spatio-temporal dynamics jointly. While another concurrent work (Watson et al., 2024) uses a transformer-based architecture for space and time, it does not tackle text-based generation for dynamic scenes but focuses on novel view synthesis from an input image. In our work, we design an efficient mechanism that enables camera control in video diffusion transformers using a ControlNet inspired mechanism without sacrificing visual quality.

3 METHOD

3.1 LARGE TEXT-TO-VIDEO TRANSFORMERS

Text-to-video generation. Diffusion models have emerged as the dominant paradigm for large-scale video generation (Ho et al., 2022b;a; Brooks et al., 2024). The standard setup considers the conditional distribution $p(\boldsymbol{x}|\boldsymbol{y})$ of videos $\boldsymbol{x} \in \mathbb{R}^{F\times H\times W}$ (consisting of F frames of $H \times W$ resolution) given their text descriptions $\boldsymbol{y} \in \mathcal{Y}^L$, consisting of L (possibly padded) tokens from the alphabet $\mathcal{Y}$. Following (Karras et al., 2022), our video diffusion framework assumes a denoising model $D_{\boldsymbol{\theta}} : (\tilde{\boldsymbol{x}}; \boldsymbol{y}, \sigma) \mapsto \hat{\boldsymbol{x}}$ that predicts a clean video $\hat{\boldsymbol{x}}$ from the corresponding noised input $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \sigma\boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ is standard Gaussian noise and $\sigma \sim \log\mathcal{N}(P_{\text{mean}}, P_{\text{std}}^2)$ is the noise strength, sampled from log-normal distribution with location P_{mean} and scale P_{std}. The model is parametrized by a neural network $F_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}; \boldsymbol{y}, \sigma)$ as $D_{\boldsymbol{\theta}}(\tilde{\boldsymbol{x}}; \boldsymbol{y}, \sigma) = c_{\text{out}}(\sigma)F_{\boldsymbol{\theta}}(c_{\text{in}}(\sigma)\tilde{\boldsymbol{x}}; \boldsymbol{y}, \sigma) + c_{\text{skip}}(\sigma)\tilde{\boldsymbol{x}}$, where $c_{\text{in}}(\sigma), c_{\text{out}}(\sigma)$ and $c_{\text{skip}}(\sigma)$ are input, output and residual scaling factors from (Karras et al., 2022). The minimization objective is defined as

$$\mathcal{L}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{p(\boldsymbol{x},\boldsymbol{y},\sigma,\boldsymbol{\varepsilon})} \left[\| D_{\boldsymbol{\theta}}(\boldsymbol{x} + \sigma\boldsymbol{\varepsilon}; \boldsymbol{y}, \sigma) - \boldsymbol{x} \|_2^2 \right]. \tag{1}$$

We refer the reader to (Menapace et al., 2024) and (Karras et al., 2022) for further details on the diffusion setup, which we adopted without modifications.

Spatiotemporal transformers. Following SnapVideo (Menapace et al., 2024), our video generator consists of two models: the base 4B-parameters generator, operating on 16-frames 36×64 resolution videos, and a 288×512 upsampler. The latter, a diffusion model itself, is fine-tuned from the base model and conditioned on the generated low-resolution videos. Each model uses FIT transformer blocks (Chen & Li, 2023; Jabri et al., 2022) for efficient self-attention operations (see Fig. 3). An FIT model consists of B blocks (we have $B = 6$ in all the experiments) and first partitions each frame in an input video into patches (Dosovitskiy et al., 2021) of resolution $h_p \times w_p$ (we use $h_p = w_p = 4$ in all the experiments). These video patches are then independently projected via a feedforward (FF) layer to obtain a sequence of video tokens $\boldsymbol{v}_{\ell:L}^{(b)} \triangleq (\boldsymbol{v}_1, ..., \boldsymbol{v}_L) \in \mathbb{R}^{L\times d}$ of length

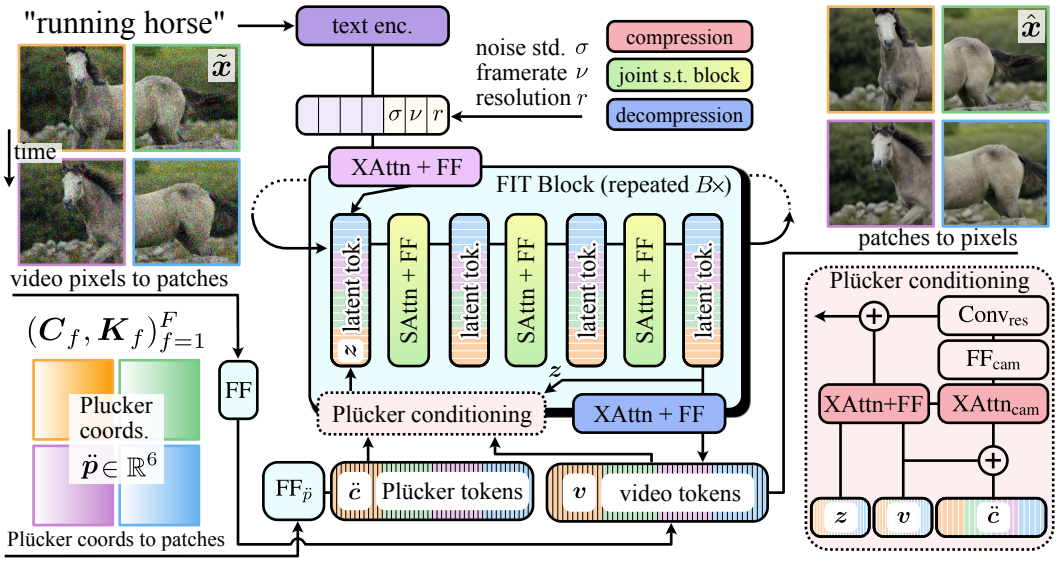

Figure 3: **Overview of architecture.** We adapt a FIT-based architecture (Chen & Li, 2023) to incorporate camera control. We take as input the noisy input video $\tilde{\boldsymbol{x}}$, camera extrinsics $\boldsymbol{C}_f$, and camera intrinsics $\boldsymbol{K}_f$ for each video frame f. We compute the Plücker coordinates for each pixel within the video frames using the camera parameters. Both the input video and Plücker coordinate frames are converted to patch tokens, and we condition the video patch tokens using a mechanism similar to ControlNet (Zhang et al., 2023) ("Plücker conditioning" block). Then, the model estimates the denoised video $\hat{\boldsymbol{x}}$ by recurrent application of FIT blocks (Chen & Li, 2023). Each block reads information from the patch tokens into a small set of latent tokens on which computation is performed. The results are written to the patch tokens in an iterative denoising diffusion process.

$L = F \times (H/h_p) \times (W/w_p)$ and dimensionality d. Next, each FIT block "reads" the information from this video sequence into a much shorter sequence of M *latent* tokens $\boldsymbol{z}_{m:M}^{(b)} \triangleq (\boldsymbol{z}_1, ..., \boldsymbol{z}_M)$ through a "`read`" cross-attention layer, followed by a feedforward layer. The core processing with self-attention layers is performed in this latent space, and then the result is written back to the video tokens through a corresponding "`write`" cross-attention layer (also followed by an FF layer). The latent tokens in each subsequent FIT block are initialized from the previous one, which helps to propagate the computational results throughout the network. In this way, the entire computation occurs jointly in both spatial and temporal axes, which yields superior scalability (Menapace et al., 2024). However, it abandons the decomposed spatial/temporal computation nature of modern video diffusion U-Nets, which modern camera conditioning techniques (Wang et al., 2023e; Yang et al., 2024b) rely on to enforce control without compromising visual quality.

3.2 Camera control for spatiotemporal transformers

Spatiotemporal camera representation. The standard way of representing camera parameters for a video (in the pinhole model) is via a trajectory of extrinsics and intrinsics camera parameters $(\boldsymbol{C}_f, \boldsymbol{K}_f)_{f=1}^F$ for each f-th frame. The matrix $\boldsymbol{C}_f = [\boldsymbol{R}; \boldsymbol{t}] \in \mathbb{R}^{3\times4}$, describes the camera rotation $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ and translation $\boldsymbol{t} \times \mathbb{R}^3$, and $\boldsymbol{K}_f \in \mathbb{R}^{3\times3}$ contains the focal length and principal point (and also horizontal/vertical skew coefficient, but it is always 0 in our setup). To control camera motion, existing methods condition the temporal attention layers of U-Net-based video generators on embeddings computed from these camera parameters (Wang et al., 2023e; Yang et al., 2024b; Hu et al., 2024). Such a pipeline provides a good conditioning signal for convolutional video generators with decomposed spatial/temporal computation, but our experiments demonstrate that it works poorly for spatiotemporal transformers: they either fail to pick up any controllability (when being added as transformed residuals to the latent tokens), or degrade the visual quality of the output (when the original network parameters are being fine-tuned). This motivates us to design a better camera conditioning scheme, tailored for modern large-scale spatiotemporal transformers.

First, we propose to normalize the camera parameters w.r.t the first frame. For this, we recompute the rotations and translations for each f-th frame as $\boldsymbol{R}'_f = \boldsymbol{R}_1^{-1}\boldsymbol{R}_f$ and $\boldsymbol{t}'_f = \boldsymbol{t}_f - \boldsymbol{t}_1$. This procedure results in normalized camera extrinsics as $\boldsymbol{C}_f{}' = [\boldsymbol{R}'_f; \boldsymbol{t}'_f]$ and establishes a consistent coordinate system across different samples in the dataset. After that, we found it essential to enrich the conditioning information by switching from temporal frame-level camera parameters to pixel-wise *spatiotemporal* ones. This is achieved by computing the Plücker coordinates for each pixel, providing a fine-grained positional representation.

Plücker coordinates provide a convenient parametrization of lines in the 3D space, and we use them to compute fine-grained positional representations of each pixel in each frame of a video. Given the extrinsic and intrinsic camera parameters $\boldsymbol{R}'_f, \boldsymbol{t}'_f, \boldsymbol{K}_f$ of the f-th frame, we parametrize each (h, w)-th pixel as a Plücker embedding $\ddot{\boldsymbol{p}}_{f,h,w} \in \mathbb{R}^6$ from the camera position to the pixel's center as

$$\ddot{\boldsymbol{p}}_{f,h,w} = (\boldsymbol{t}_f{}' \times \hat{\boldsymbol{d}}_{f,h,w}, \hat{\boldsymbol{d}}_{f,h,w}), \quad \hat{\boldsymbol{d}}_{f,h,w} = \frac{\boldsymbol{d}}{\|\boldsymbol{d}_{f,h,w}\|}, \quad \boldsymbol{d}_{f,h,w} = \boldsymbol{R}_f{}'\boldsymbol{K}_f[w, h, 1]^\top + \boldsymbol{t}_f{}'. \quad (2)$$

This approach mirrors the technique used in recent 3D works (Sitzmann et al., 2021; Chen et al., 2023a; Kant et al., 2024) as well as CameraCtrl (He et al., 2024a), a concurrent study focusing on camera control in U-Net-based video diffusion models. The motivation for using Plücker coordinates is that geometric manipulations in the Plücker space can be performed through simple arithmetics on the coordinates, which makes it easier for the network to use the positional information stored in such a disentangled representation.

Computing Plücker coordinates for each pixel results in a $\ddot{\boldsymbol{P}} \in \mathbb{R}^{6\times F\times H\times W}$ spatiotemporal camera representation for a video. To input it into the model, we first perform the equivalent ViT-like (Dosovitskiy et al., 2021) $h_p \times w_p$ patchification procedure. It is followed by a learnable 2-layered MLP $\text{MLP}_{\ddot{p}}$ with a GELU (Hendrycks & Gimpel, 2016) non-linearity to obtain the Plücker camera tokens sequence $\ddot{\boldsymbol{c}}_{\ell:L} \in \mathbb{R}^{L\times d}$ of the same length $L = F \times (H/h_p) \times (W/w_p)$ and dimensionality d as the video tokens sequence $\boldsymbol{v}^{(b)}_{\ell:L}$. This spatiotemporal representation carries fine-grained positional information about each pixel in a video, making it easier for the generator to accurately follow the desired camera motion.

Camera conditioning. To input the Plücker embeddings into our video generator, we design an efficient ControlNet like (Zhang et al., 2023) mechanism tailored for large transformer models (see Fig. 3). This mechanism is guided by two main objectives: 1) the model should be amenable to rapid fine-tuning from a small dataset with estimated camera positions; and 2) the visual quality shouldn't be compromised during the fine-tuning stage. We found that meeting these objectives is more challenging for spatiotemporal transformers compared to U-Net-based models with decomposed spatial/temporal computation, since even minor interventions into their design quickly lead to degraded video outputs. We hypothesize that the core reason for it is the entangled spatial/temporal computation of video transformers: any attempt to alter the temporal dynamics (such as camera motion) influences spatial communication between the tokens, leading to unnecessary signal propagation and overfitting during the fine-tuning stage. To mitigate this, we input the camera information gradually through `read` cross-attention layers, zero-initialized from the original network parameters of the corresponding layers.

Specifically, in each b-th FIT block of our video generator, we replace its standard `read` cross-attention operation (see (Jabri et al., 2022; Menapace et al., 2024) for details):

$$\boldsymbol{z}'^{(b)}_{m:M} = \text{FF}^{(b)}(\text{XAttn}^{(b)}(\boldsymbol{z}^{(b)}_{m:M}, \boldsymbol{v}^{(b)}_{\ell:L})), \quad (3)$$

where $\text{FF}(\cdot)$ and $\text{XAttn}(\cdot,\cdot)$ denote feed-forward and cross-attention layers respectively. Our revised formulation is given as:

$$\boldsymbol{z}'^{(b)}_{m:M} = \text{FF}^{(b)}(\text{XAttn}^{(b)}(\boldsymbol{z}^{(b)}_{m:M}, \boldsymbol{v}^{(b)}_{\ell:L})) + \text{Conv}^{(b)}_{\text{res}}\left[\text{FF}^{(b)}_{\text{cam}}(\text{XAttn}^{(b)}_{\text{cam}}(\boldsymbol{z}^{(b)}_{m:M}, \ddot{\boldsymbol{c}}_{\ell:L} + \boldsymbol{v}^{(b)}_{\ell:L}))\right], \quad (4)$$

where $\text{FF}^{(b)}_{\text{cam}}$, and $\text{XAttn}^{(b)}_{\text{cam}}$ are learnable layers, $\text{Conv}^{(b)}_{\text{res}}$ is a 1-dimensional convolution that processes camera-augmented latents. The produced latents $\boldsymbol{z}'^{(b)}_{m:M}$ are then passed to the sequence of four self-attention layers, which are the core computational component of FIT. It is crucial to instantiate the weights of the output convolutions $\text{Conv}^{(b)}_{\text{res}}$ from zeros to preserve the model initialization. Besides, we initialize the weights of $\text{FF}^{(b)}_{\text{cam}}$ and $\text{XAttn}^{(b)}_{\text{cam}}$ from the corresponding parameters of the original

Figure 4: **Camera-conditioned text-to-video generation.** Comparison of the proposed approach to MotionCtrl and CameraCtrl for the same camera trajectory input. We adapted MotionCtrl and CameraCtrl to the same transformer-based model for fair comparisons. MotionCtrl exhibits worse quality due to fine-tuning existing layers, and CameraCtrl is not sensitive to camera conditioning. See the supplementary webpage for video results.

network. This approach helps to preserve visual quality at initialization and facilitates rapid fine-tuning on a small dataset. As a result, we obtain the method for fine-grained 3D camera control in large video diffusion transformers. We name it VD3D and visualize its architecture in Figure 3.

3.3 Training details

To ensure comparability with prior work such as (Wang et al., 2023e), we train our video generator on the same RealEstate10k dataset (Zhou et al., 2018). We optimize only the newly added parameters $\text{FF}_{\ddot{p}}$ and $(\text{FF}^{(b)}, \text{XAttn}^{(b)})_{b=1}^{B}$, and keep the rest of the network frozen. We found that training only the base 36×64 model was sufficient, as the 288×512 upsampler already accurately follows the camera motion of a low-resolution video.

3.4 Dataset

We fine-tune a text-to-video model, pre-trained on 2D video data, on RealEstate10K (Zhou et al., 2018). The training split for fine-tuning consists of roughly 65K video clips, and is the same as is used in concurrent work (MotionCtrl (Wang et al., 2023e) and CameraCtrl (He et al., 2024a)).

3.5 Metrics

We conduct a user study to evaluate our approach for camera-controlled text-to-video generation. The study participants are presented with 20 side-by-side comparisons between the proposed approach and the baselines as well as a reference video from RealEstate10K with the same trajectory to better judge the camera alignment. We ask 20 participants for each generated video sequence to indicate which generated video they prefer based on multiple submetrics, namely, camera alignment, motion quality, text alignment, visual quality, and overall preference. The user study involved negligible risk to the participants and was conducted with appropriate institutional review board and legal approval. We evaluate our method using 20 camera trajectories sampled from the RealEstate10K test split that were not seen during training for the user study. We use the full test split combined with unseen text prompts for the automated camera evaluations, i.e., 6928 unseen camera trajectories combined with out-of-distribution text prompts.

3.6 Baselines

We compare our work to MotionCtrl (Wang et al., 2023e) and the concurrent work CameraCtrl (He et al., 2024a) by adapting their publicly released code to the same pre-trained video model as ours. Note that both baselines were originally designed for space-time disentangled U-Net video diffusion models. Hence, their approaches are not directly applicable to the spatio-temporal transformers, and so we adapt them to this setting as follows. For MotionCtrl, we omit their object motion control

module and use their proposed camera motion control module to encode the camera parameters into a context vector used with the SnapVideo model. We fine-tune both the camera motion control module and the cross attention between the latent vectors and the patches. For CameraCtrl, we fine-tune the original camera encoder module and use this to produce the latent vectors in the SnapVideo model. During fine-tuning the model weights are kept frozen—i.e., the same as in our proposed approach. Furthermore, we include comparisons to the original MotionCtrl and CameraCtrl works built on the U-Net-based AnimateDiff (Guo et al., 2024) architecture. For fair comparison, we evaluate all metrics for MotionCtrl (U-Net) and CameraCtrl (U-Net) on the same test sets as our method using the publicly provided code and checkpoints. Moreover, we provide metrics for the base model used across all our experiments, i.e., a model without camera injection. We provide more baseline variants of MotionCtrl and CameraCtrl in the appendix in Sec. B.3.

4 EXPERIMENTS

4.1 ASSESSMENT

We provide a qualitative and quantitative assessment of our approach compared to the baselines in Fig. 4 and in Tab. 1. Following CameraCtrl (He et al., 2024a), we also evaluate the camera pose accuracy using ParticleSfM (Zhao et al., 2022) on generated videos in Tab. 2. We use generations for text prompts from RealEstate10K (Zhou et al., 2018) and MSR-VTT (Xu et al., 2016), testing both in- and out-of-distribution prompts. Note that we adjust the CameraCtrl (He et al., 2024a) evaluation pipeline by normalizing all cameras into a unified scale as COLMAP provides different scales across different scenes. This prevents scenes with large scale to have a higher impact on the errors. Please also refer to the supplementary webpage for additional video results.

In Fig. 4 we observe that adapting the camera conditioning method from the MotionCtrl degrades visual quality and text alignment, likely because this approach adjusts the weights of the base video model. In the space-time U-Net for which this approach was proposed, the temporal layers can be fine-tuned without sacrificing visual fidelity. Since spatio-temporal transformers do not decompose temporal and spatial attributes in the same way, the model overfits to the small dataset used to fine-tune the cross-attention layer. While we observe some agreement with the camera poses used to condition the model, the text alignment is generally low in our experiments (see supplemental webpage). In contrast, CameraCtrl keeps the pre-trained video model weights frozen and only trains a camera encoder. This leads to strong visual quality, but the generated videos show little agreement with the input camera poses. For fair comparison, we trained all models for the same number of iterations (described in Sec. 3.3).

The results of the user study in Tab. 1 show that most participants prefer the generated videos using the proposed camera conditioning mechanism across all evaluated sub-metrics. We also observe a pronounced preference for the camera alignment of the proposed method compared to the other baselines. That is, 82% and 78% of participants prefer the camera alignment of the proposed method compared to our respective adaptations of MotionCtrl and CameraCtrl to the video transformer model. All results are significant at the $p < 0.001$ level as evaluated using a χ^2 test. We further present image-based metrics for multi-view generation and video generation quality metrics in the appendix in Sec. B.1 and Tab. B.2 respectively. The results of our camera pose accuracy evaluation in Tab. 2 further demonstrate that our approach clearly outperforms previous works. Note that the previous works MotionCtrl and CameraCtrl especially struggle with the rotation accuracy.

4.2 ABLATIONS

Plücker embedding. We motivate our Plücker embedding conditioning mechanism by training a variant using the raw camera matrices. Concretely, we flatten and concatenate extrinsics and intrinsics matrices in the channel dimension and repeat the values in the spatial patch dimensions. We observe that Plücker embeddings provide an essential spatial conditioning mechanism, as shown in Tab. 2.

ControlNet conditioning. Our ControlNet-inspired conditioning mechanism ensures fast and precise learning of the conditioning signal distribution. Instead of using a ControlNet block, we simply add zero-initialized Plücker embedding features to the patches and observe close to no camera control. We observe training cross-attention layers in the ControlNet block is key to learning camera control while preserving the original model weights. This is confirmed by our camera evaluation in Tab. 2.

Table 1: **Quantitative results.** We compare our method to MotionCtrl and CameraCtrl implemented on the same base video models as ours. The methods are evaluated in a user study in which participants indicate their preference based on camera alignment (CA), motion quality (MQ), text alignment (TA), visual quality (VQ), and overall preference (Overall). The percentages indicate preference for VD3D vs. the alternative method (in each row). All results are statistically significant with $p < 0.001$ as evaluated using a χ^2 test.

	Human Preference				
Method	CA	MQ	TA	VQ	Overall
VD3D vs. MotionCtrl	82%	81%	86%	81%	84%
VD3D vs. CameraCtrl	78%	64%	63%	65%	66%

Table 2: **Camera pose evaluation.** We evaluate all models using reference camera trajectories from the RealEstate10K test set. We compute translation and rotation errors based on estimated camera poses from generations using ParticleSfM (Zhao et al., 2022).

Method	RealEstate10K		MSR-VTT	
	TransError (↓)	RotError (↓)	TransError (↓)	RotError (↓)
Base Model	0.616	0.207	0.717	0.216
MotionCtrl (U-Net)	0.477	0.094	0.593	0.137
CameraCtrl (U-Net)	0.465	0.089	0.587	0.132
MotionCtrl	0.518	0.161	0.627	0.148
CameraCtrl	0.532	0.165	0.578	0.220
Ours	**0.409**	**0.043**	**0.504**	**0.050**
w/o Plucker	0.517	0.161	0.676	0.156
w/o ControlNet	0.573	0.182	0.787	0.179
w/o weight copy	0.424	0.044	0.513	0.063
w/o add context	0.602	0.212	0.702	0.128
w/o Plucker context	0.487	0.088	0.627	0.091

ControlNet weight copy. While the ControlNet block is essential, copying the pre-trained weights into the new copy has rather minor impact, as shown in Tab. 2. To verify this, we train a model where we randomly initialize the cross-attention block between patches and latents instead of copying the weights. We observe similar results, showing that rather the architecture and zero-initialization are the key component of this design.

Add to context vector. We train a model where we use camera matrices, linearly encode them, and add them to the context vector as a simple conditioning mechanism for transformer-based models. The rendered videos show little to no correlation with the input camera matrices, resulting in poor camera pose accuracy, as shown in Tab. 2. This highlights the importance of carefully incorporating cameras into the spatio-temporal patches with our Plücker embedding block.

Plücker in context vector. We train a model where we integrate Plücker embeddings into the context vector instead of the patches. We similarly compute Plücker embeddings, but instead of patchifying them, we flatten and linearly map the embeddings into a vector that has the same shape as the context vector. We fuse the Plücker features and context vector using the same Plücker conditioning mechanism used for the patches. We observe higher camera pose errors in Tab. 2, highlighting that it is crucial to incorporate the spatio-temporal Plücker embeddings into the spatio-temporal patches instead of using the context vector.

4.3 Applications

Image-to-video generation. We synthesize camera-controlled videos based on different camera poses as shown in Fig. 1. For this task we use a version of our pre-trained text-to-video model that we fine-tune on video sequences where a random subset of the input frames are masked. At

Figure 5: **Conditional multi-view generation on a real image.** We can generate arbitrary camera trajectories from a given real image for multi-view synthesis, paving the way to single-image scene reconstruction using camera-controlled video models.

inference time, we can provide image guidance for any of the generated frames, providing an additional dimension of controllability when paired with our proposed method for camera control. To demonstrate image-to-video generation in Fig. 1, we condition the model using camera poses along with image guidance from the first frame of a generated video sequence. Note that our method provides no control over motion within the scene itself; hence, scene motion can differ depending on the random seed or the provided camera poses.

Image-to-multiview generation. We also explore multi-view generation for static scenes as shown in Fig. 5. Given a real input image of a complex scene unseen during training, our camera-conditioned model generates view-consistent renderings of that scene from arbitrary viewpoints. These multi-view renderings could be directly relevant to downstream 3D reconstructions pipelines, e.g., based on NeRF (Mildenhall et al., 2020) or 3D Gaussian Splatting (Kerbl et al., 2023). We show the potential of camera-conditioned image-to-multiview generation for complex 3D scene generation, but we leave more extensive exploration of this topic for future work.

5 Conclusion

Large-scale video transformer models show immense promise to solve many long-standing challenges in computer vision, including novel-view synthesis, single-image 3D reconstruction, and text-conditioned scene synthesis. Our work brings additional controllability to these models, enabling a user to specify the camera poses from which video frames are rendered.

Limitations and future work. There are several limitations to our work, which highlight important future research directions. For example, while rendering static scenes from different camera viewpoints produces results that appear 3D consistent, dynamic scenes rendered from different camera viewpoints can have inconsistent motion (see supplemental videos). We envision that future video generation models will have fine-grained control over both scene motion and camera motion to address this issue. Further, our approach applies camera conditioning only to the low-resolution SnapVideo model and we keep their upsampler model frozen (i.e., without camera conditioning)—it may be possible to further improve camera control through joint training, though this brings additional architectural engineering and computational challenges. Finally, our approach is currently limited to generation of relatively short videos (16 frames), based on the design and training scheme of the SnapVideo model. Future work to address these limitations will enable new capabilities for applications in computer vision, visual effects, augmented and virtual reality, and beyond.

Broader impacts. Recent video generation models demonstrate coherent, photorealistic synthesis of complex scenes—capabilities that are highly sought after for numerous applications across computer vision, graphics, and beyond. Our key technical contributions relate to camera control of these models, which can be applied to a wide range of methods. As with all generative models and technologies, underlying technologies can be misused by bad actors in unintended ways. While these methods continue to improve, researchers and developers should continue to consider safeguards, such as output filtering, watermarking, access control, and others.

6 Reproducibility Statement

We have structured our paper to ensure comprehensive reproducibility of our camera control method. Section 3 provides a detailed description of our method, including theoretical foundations and core algorithmic components. Section 4 presents thorough experimental details, covering evaluation protocols, metrics, and comparisons with baselines. All implementation specifics, including training procedures, architectural choices, and hyperparameters, are documented in Appendix C. To further facilitate reproducibility, we include the source code of our camera-controlled FIT block as supplementary material. The provided implementation contains the core components necessary to replicate our approach, accompanied by documentation and usage examples. We welcome requests for additional technical details or clarifications to support the research community in reproducing and building upon our work.

References

Sherwin Bahmani, Jeong Joon Park, Despoina Paschalidou, Hao Tang, Gordon Wetzstein, Leonidas Guibas, Luc Van Gool, and Radu Timofte. 3d-aware video generation. In *TMLR*, 2023a.

Sherwin Bahmani, Jeong Joon Park, Despoina Paschalidou, Xingguang Yan, Gordon Wetzstein, Leonidas Guibas, and Andrea Tagliasacchi. CC3D: Layout-conditioned generation of compositional 3D scenes. *Proc. ICCV*, 2023b.

Sherwin Bahmani, Xian Liu, Wang Yifan, Ivan Skorokhodov, Victor Rong, Ziwei Liu, Xihui Liu, Jeong Joon Park, Sergey Tulyakov, Gordon Wetzstein, Andrea Tagliasacchi, and David B. Lindell. Tc4d: Trajectory-conditioned text-to-4d generation. In *Proc. ECCV*, 2024a.

Sherwin Bahmani, Ivan Skorokhodov, Victor Rong, Gordon Wetzstein, Leonidas Guibas, Peter Wonka, Sergey Tulyakov, Jeong Joon Park, Andrea Tagliasacchi, and David B. Lindell. 4d-fy: Text-to-4d generation using hybrid score distillation sampling. In *Proc. CVPR*, 2024b.

Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *Proc. ICCV*, 2021.

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 *Proc. CVPR*, 2023b.

Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. *OpenAI technical reports*, 2024. URL `https://openai.com/research/video-generation-models-as-world-simulators`.

Zenghao Chai, Chen Tang, Yongkang Wong, and Mohan Kankanhalli. Star: Skeleton-aware text-based 4d avatar generation with in-network motion retargeting. *arXiv preprint arXiv:2406.04629*, 2024.

Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3D generative adversarial networks. *Proc. CVPR*, 2022.

Eric R Chan, Koki Nagano, Matthew A Chan, Alexander W Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. Generative novel view synthesis with 3d-aware diffusion models. In *Proc. ICCV*, 2023.

Eric Ming Chen, Sidhanth Holalkere, Ruyu Yan, Kai Zhang, and Abe Davis. Ray conditioning: Trading photo-consistency for photo-realism in multi-view image generation. In *Proc. ICCV*, 2023a.

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

Junsong Chen, Yue Wu, Simian Luo, Enze Xie, Sayak Paul, Ping Luo, Hang Zhao, and Zhenguo Li. Pixart-{\delta}: Fast and controllable image generation with latent consistency models. *arXiv preprint arXiv:2401.05252*, 2024.

Kevin Chen, Christopher B Choy, Manolis Savva, Angel X Chang, Thomas Funkhouser, and Silvio Savarese. Text2Shape: Generating shapes from natural language by learning joint embeddings. *Proc. ACCV*, 2018.

Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. Fantasia3D: Disentangling geometry and appearance for high-quality text-to-3D content creation. *arXiv preprint arXiv:2303.13873*, 2023c.

Ting Chen and Lala Li. Fit: Far-reaching interleaved transformers. *arXiv preprint arXiv:2305.12689*, 2023.

Wen-Hsuan Chu, Lei Ke, and Katerina Fragkiadaki. Dreamscene4d: Dynamic multi-object scene generation from monocular videos. *arXiv preprint arXiv:2405.02280*, 2024.

Terrance DeVries, Miguel Angel Bautista, Nitish Srivastava, Graham W Taylor, and Joshua M Susskind. Unconstrained scene generation with locally conditioned radiance fields. *Proc. ICCV*, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *Proc. ICLR*, 2021.

Qijun Feng, Zhen Xing, Zuxuan Wu, and Yu-Gang Jiang. FDGaussian: Fast Gaussian splatting from single image via geometric-aware diffusion model. *arXiv preprint arXiv:2403.10242*, 2024a.

Yutao Feng, Yintong Shang, Xiang Feng, Lei Lan, Shandian Zhe, Tianjia Shao, Hongzhi Wu, Kun Zhou, Hao Su, Chenfanfu Jiang, et al. Elastogen: 4d generative elastodynamics. *arXiv preprint arXiv:2405.15056*, 2024b.

Peng Gao, Le Zhuo, Ziyi Lin, Dongyang Liu, Ruoyi Du, Xu Luo, Longtian Qiu, Yuhang Zhang, et al. Lumina-t2x: Transforming text into any modality, resolution, and duration via flow-based large diffusion transformers. *arXiv preprint arXiv:2405.05945*, 2024a.

Quankai Gao, Qiangeng Xu, Zhe Cao, Ben Mildenhall, Wenchao Ma, Le Chen, Danhang Tang, and Ulrich Neumann. Gaussianflow: Splatting Gaussian dynamics for 4D content creation. *arXiv preprint arXiv:2403.12365*, 2024b.

Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024c.

William Gao, Noam Aigerman, Thibault Groueix, Vova Kim, and Rana Hanocka. TextDeformer: Geometry manipulation using text guidance. *Proc. SIGGRAPH*, 2023.

Jiatao Gu, Alex Trevithick, Kai-En Lin, Joshua M Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. NerfDiff: Single-image view synthesis with NeRF-guided distillation from 3D-aware diffusion. *Proc. ICML*, 2023.

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

Junlin Han, Filippos Kokkinos, and Philip Torr. VFusion3D: Learning scalable 3D generative models from video diffusion models. *arXiv preprint arXiv:2403.12034*, 2024.

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*, 2024a.

Xianglong He, Junyi Chen, Sida Peng, Di Huang, Yangguang Li, Xiaoshui Huang, Chun Yuan, Wanli Ouyang, and Tong He. GVGEN: Text-to-3D generation with volumetric representation. *arXiv preprint arXiv:2403.12957*, 2024b.

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

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

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. *Proc. NeurIPS*, 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*, 2022a.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Proc. NeurIPS*, 2022b.

Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. In *Proc. ICCV*, 2023.

Lukas Höllein, Aljaž Božič, Norman Müller, David Novotny, Hung-Yu Tseng, Christian Richardt, Michael Zollhöfer, and Matthias Nießner. ViewDiff: 3D-consistent image generation with text-to-image models. *Proc. CVPR*, 2024.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3D. *Proc. ICLR*, 2024.

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

Teng Hu, Jiangning Zhang, Ran Yi, Yating Wang, Hongrui Huang, Jieyu Weng, Yabiao Wang, and Lizhuang Ma. Motionmaster: Training-free camera motion transfer for video generation. *arXiv preprint arXiv:2404.15789*, 2024.

Tianyu Huang, Yihan Zeng, Hui Li, Wangmeng Zuo, and Rynson WH Lau. Dreamphysics: Learning physical properties of dynamic 3d gaussians with video diffusion priors. *arXiv preprint arXiv:2406.01476*, 2024.

Allan Jabri, David Fleet, and Ting Chen. Scalable adaptive computation for iterative generation. *arXiv preprint arXiv:2212.11972*, 2022.

Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. *Proc. CVPR*, 2022.

Nikolay Jetchev. ClipMatrix: Text-controlled creation of 3D textured meshes. *arXiv preprint arXiv:2109.12922*, 2021.

Lutao Jiang and Lin Wang. Brightdreamer: Generic 3D Gaussian generative framework for fast text-to-3D synthesis. *arXiv preprint arXiv:2403.11273*, 2024.

Yanqin Jiang, Li Zhang, Jin Gao, Weimin Hu, and Yao Yao. Consistent4d: Consistent 360 {\deg} dynamic object generation from monocular video. *Proc. ICLR*, 2024.

Yash Kant, Aliaksandr Siarohin, Ziyi Wu, Michael Vasilkovsky, Guocheng Qian, Jian Ren, Riza Alp Guler, Bernard Ghanem, Sergey Tulyakov, and Igor Gilitschenski. Spad: Spatially aware multi-view diffusers. In *Proc. CVPR*, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Proc. NeurIPS*, 2022.

Oren Katzir, Or Patashnik, Daniel Cohen-Or, and Dani Lischinski. Noise-free score distillation. *Proc. ICLR*, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 2023.

Seung Wook Kim, Bradley Brown, Kangxue Yin, Karsten Kreis, Katja Schwarz, Daiqing Li, Robin Rombach, Antonio Torralba, and Sanja Fidler. NeuralField-LDM: Scene generation with hierarchical latent diffusion models. *Proc. CVPR*, 2023.

Zhengfei Kuang, Shengqu Cai, Hao He, Yinghao Xu, Hongsheng Li, Leonidas Guibas, and Gordon Wetzstein. Collaborative video diffusion: Consistent multi-video generation with camera control. *arXiv preprint arXiv:2405.17414*, 2024.

Nupur Kumari, Grace Su, Richard Zhang, Taesung Park, Eli Shechtman, and Jun-Yan Zhu. Customizing text-to-image diffusion with camera viewpoint control. *arXiv preprint arXiv:2404.12333*, 2024.

Kyungmin Lee, Kihyuk Sohn, and Jinwoo Shin. DreamFlow: High-quality text-to-3D generation by approximating probability flow. *Proc. ICLR*, 2024a.

Yao-Chih Lee, Yi-Ting Chen, Andrew Wang, Ting-Hsuan Liao, Brandon Y Feng, and Jia-Bin Huang. Vividdream: Generating 3d scene with ambient dynamics. *arXiv preprint arXiv:2405.20334*, 2024b.

Bing Li, Cheng Zheng, Wenxuan Zhu, Jinjie Mai, Biao Zhang, Peter Wonka, and Bernard Ghanem. Vivid-zoo: Multi-view video generation with diffusion model. *arXiv preprint arXiv:2406.08659*, 2024a.

Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. Instant3D: Fast text-to-3D with sparse-view generation and large reconstruction model. *Proc. ICLR*, 2024b.

Renjie Li, Panwang Pan, Bangbang Yang, Dejia Xu, Shijie Zhou, Xuanyang Zhang, Zeming Li, Achuta Kadambi, Zhangyang Wang, and Zhiwen Fan. 4k4dgen: Panoramic 4d generation at 4k resolution. *arXiv preprint arXiv:2406.13527*, 2024c.

Zhiqi Li, Yiming Chen, Lingzhe Zhao, and Peidong Liu. Controllable text-to-3D generation via surface-aligned Gaussian splatting. *arXiv preprint arXiv:2403.09981*, 2024d.

Hanwen Liang, Yuyang Yin, Dejia Xu, Hanxue Liang, Zhangyang Wang, Konstantinos N Plataniotis, Yao Zhao, and Yunchao Wei. Diffusion4d: Fast spatial-temporal consistent 4d generation via video diffusion models. *arXiv preprint arXiv:2405.16645*, 2024.

Yixun Liang, Xin Yang, Jiantao Lin, Haodong Li, Xiaogang Xu, and Yingcong Chen. Luciddreamer: Towards high-fidelity text-to-3D generation via interval score matching. *arXiv preprint arXiv:2311.11284*, 2023.

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3D: High-resolution text-to-3D content creation. *Proc. CVPR*, 2023a.

Yukang Lin, Haonan Han, Chaoqun Gong, Zunnan Xu, Yachao Zhang, and Xiu Li. Consistent123: One image to highly consistent 3D asset using case-aware diffusion priors. *arXiv preprint arXiv:2309.17261*, 2023b.

Huan Ling, Seung Wook Kim, Antonio Torralba, Sanja Fidler, and Karsten Kreis. Align your gaussians: Text-to-4d with dynamic 3d gaussians and composed diffusion models. In *Proc. CVPR*, 2024a.

Pengyang Ling, Jiazi Bu, Pan Zhang, Xiaoyi Dong, Yuhang Zang, Tong Wu, Huaian Chen, Jiaqi Wang, and Yi Jin. Motionclone: Training-free motion cloning for controllable video generation. *arXiv preprint arXiv:2406.05338*, 2024b.

Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Proc. NeurIPS*, 2024a.

Pengkun Liu, Yikai Wang, Fuchun Sun, Jiafang Li, Hang Xiao, Hongxiang Xue, and Xinzhou Wang. Isotropic3D: Image-to-3D generation based on a single clip embedding. *arXiv preprint arXiv:2403.10395*, 2024b.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3D object. *Proc. ICCV*, 2023.

Xian Liu, Xiaohang Zhan, Jiaxiang Tang, Ying Shan, Gang Zeng, Dahua Lin, Xihui Liu, and Ziwei Liu. HumanGaussian: Text-driven 3D human generation with Gaussian splatting. *Proc. CVPR*, 2024c.

Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. SyncDreamer: Generating multiview-consistent images from a single-view image. *Proc. ICLR*, 2024d.

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3D: Single image to 3D using cross-domain diffusion. *Proc. CVPR*, 2024.

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

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

Willi Menapace, Aliaksandr Siarohin, Ivan Skorokhodov, Ekaterina Deyneka, Tsai-Shien Chen, Anil Kag, Yuwei Fang, Aleksei Stoliar, Elisa Ricci, Jian Ren, et al. Snap video: Scaled spatiotemporal transformers for text-to-video synthesis. *Proc. CVPR*, 2024.

Qiaowei Miao, Yawei Luo, and Yi Yang. Pla4d: Pixel-level alignments for text-to-4d gaussian splatting. *arXiv preprint arXiv:2405.19957*, 2024.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Proc. ECCV*, 2020.

Norman Müller, Katja Schwarz, Barbara Rössle, Lorenzo Porzi, Samuel Rota Bulò, Matthias Nießner, and Peter Kontschieder. Multidiff: Consistent novel view synthesis from a single image. In *Proc. CVPR*, 2024.

Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. StyleSDF: High-resolution 3D-consistent image and geometry generation. *Proc. CVPR*, 2022.

Zijie Pan, Zeyu Yang, Xiatian Zhu, and Li Zhang. Fast dynamic 3d object generation from a single-view video. *arXiv preprint arXiv:2401.08742*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. *Proc. ICCV*, 2023.

Ryan Po, Wang Yifan, Vladislav Golyanik, Kfir Aberman, Jonathan T Barron, Amit H Bermano, Eric Ryan Chan, Tali Dekel, Aleksander Holynski, Angjoo Kanazawa, et al. State of the art on diffusion models for visual computing. *arXiv preprint arXiv:2310.07204*, 2023.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D diffusion. In *Proc. ICLR*, 2023.

Guocheng Qian, Junli Cao, Aliaksandr Siarohin, Yash Kant, Chaoyang Wang, Michael Vasilkovsky, Hsin-Ying Lee, Yuwei Fang, Ivan Skorokhodov, Peiye Zhuang, et al. Atom: Amortized text-to-mesh using 2d diffusion. *arXiv preprint arXiv:2402.00867*, 2024a.

Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. *Proc. ICLR*, 2024b.

Guocheng Qian, Jinjie Mai, Abdullah Hamdi, Jian Ren, Aliaksandr Siarohin, Bing Li, Hsin-Ying Lee, Ivan Skorokhodov, Peter Wonka, Sergey Tulyakov, et al. Magic123: One image to high-quality 3D object generation using both 2D and 3D diffusion priors. *Proc. ICLR*, 2024c.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *Proc. ICML*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Proc. JMLR*, 2020.

Jiawei Ren, Liang Pan, Jiaxiang Tang, Chi Zhang, Ang Cao, Gang Zeng, and Ziwei Liu. DreamGaussian4D: Generative 4D Gaussian splatting. *arXiv preprint arXiv:2312.17142*, 2023.

Jiawei Ren, Kevin Xie, Ashkan Mirzaei, Hanxue Liang, Xiaohui Zeng, Karsten Kreis, Ziwei Liu, Antonio Torralba, Sanja Fidler, Seung Wook Kim, et al. L4gm: Large 4d gaussian reconstruction model. *arXiv preprint arXiv:2406.10324*, 2024.

Aditya Sanghi, Hang Chu, Joseph G Lambourne, Ye Wang, Chin-Yi Cheng, Marco Fumero, and Kamal Rahimi Malekshan. CLIP-Forge: Towards zero-shot text-to-shape generation. *Proc. CVPR*, 2022.

Katja Schwarz, Axel Sauer, Michael Niemeyer, Yiyi Liao, and Andreas Geiger. VoxGRAF: Fast 3D-aware image synthesis with sparse voxel grids. *Proc. NeurIPS*, 2022.

Abhishek Sharma, Adams Yu, Ali Razavi, Andeep Toor, Andrew Pierson, Ankush Gupta, Austin Waters, Daniel Tanis, Dumitru Erhan, Eric Lau, Eleni Shaw, Gabe Barth-Maron, Greg Shaw, Han Zhang, Henna Nandwani, Hernan Moraldo, Hyunjik Kim, Irina Blok, Jakob Bauer, Jeff Donahue, Junyoung Chung, Kory Mathewson, Kurtis David, Lasse Espeholt, Marc van Zee, Matt McGill, Medhini Narasimhan, Miaosen Wang, Mikołaj Bińkowski, Mohammad Babaeizadeh, Mohammad Taghi Saffar, Nick Pezzotti, Pieter-Jan Kindermans, Poorva Rane, Rachel Hornung, Robert Riachi, Ruben Villegas, Rui Qian, Sander Dieleman, Serena Zhang, Serkan Cabi, Shixin Luo, Shlomi Fruchter, Signe Nørly, Srivatsan Srinivasan, Tobias Pfaff, Tom Hume, Vikas Verma, Weizhe Hua, William Zhu, Xinchen Yan, Xinyu Wang, Yelin Kim, Yuqing Du, and Yutian Chen. Veo, 2024. URL `https://deepmind.google/technologies/veo/`.

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3D generation. *Proc. ICLR*, 2024.

Jaidev Shriram, Alex Trevithick, Lingjie Liu, and Ravi Ramamoorthi. Realmdreamer: Text-driven 3d scene generation with inpainting and depth diffusion. *arXiv preprint arXiv:2404.07199*, 2024.

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. *Proc. ICLR*, 2023a.

Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, et al. Text-to-4d dynamic scene generation. In *Proc. ICML*, 2023b.

Vincent Sitzmann, Semon Rezchikov, Bill Freeman, Josh Tenenbaum, and Fredo Durand. Light field networks: Neural scene representations with single-evaluation rendering. *Proc. NeurIPS*, 2021.

Jingxiang Sun, Bo Zhang, Ruizhi Shao, Lizhen Wang, Wen Liu, Zhenda Xie, and Yebin Liu. DreamCraft3D: Hierarchical 3D generation with bootstrapped diffusion prior. *Proc. ICLR*, 2024a.

Qi Sun, Zhiyang Guo, Ziyu Wan, Jing Nathan Yan, Shengming Yin, Wengang Zhou, Jing Liao, and Houqiang Li. Eg4d: Explicit generation of 4d object without score distillation. *arXiv preprint arXiv:2405.18132*, 2024b.

Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Viewset diffusion:(0-) image-conditioned 3d generative models from 2d data. In *Proc. ICCV*, 2023.

Stanislaw Szymanowicz, Eldar Insafutdinov, Chuanxia Zheng, Dylan Campbell, João F Henriques, Christian Rupprecht, and Andrea Vedaldi. Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. *arXiv preprint arXiv:2406.04343*, 2024a.

Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3D reconstruction. *Proc. CVPR*, 2024b.

Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. LGM: Large multi-view gaussian model for high-resolution 3d content creation. *Proc. ECCV*, 2024.

Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3D: High-fidelity 3D creation from a single image with diffusion prior. *arXiv preprint arXiv:2303.14184*, 2023.

Ayush Tewari, Tianwei Yin, George Cazenavette, Semon Rezchikov, Josh Tenenbaum, Frédo Durand, Bill Freeman, and Vincent Sitzmann. Diffusion with forward models: Solving stochastic inverse problems without direct supervision. *Proc. NeurIPS*, 2023.

Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3D object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024.

Hung-Yu Tseng, Qinbo Li, Changil Kim, Suhib Alsisan, Jia-Bin Huang, and Johannes Kopf. Consistent view synthesis with pose-guided diffusion models. In *Proc. CVPR*, 2023.

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.

Lukas Uzolas, Elmar Eisemann, and Petr Kellnhofer. Motiondreamer: Zero-shot 3d mesh animation from video diffusion models. *arXiv preprint arXiv:2405.20155*, 2024.

Basile Van Hoorick, Rundi Wu, Ege Ozguroglu, Kyle Sargent, Ruoshi Liu, Pavel Tokmakov, Achal Dave, Changxi Zheng, and Carl Vondrick. Generative camera dolly: Extreme monocular dynamic novel view synthesis. *arXiv preprint arXiv:2405.14868*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proc. NeurIPS*, 2017.

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*, 2024a.

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*, 2024b.

Ziyu Wan, Despoina Paschalidou, Ian Huang, Hongyu Liu, Bokui Shen, Xiaoyu Xiang, Jing Liao, and Leonidas Guibas. CAD: Photorealistic 3D generation via adversarial distillation. *Proc. CVPR*, 2024.

Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-NeRF: Text-and-image driven manipulation of neural radiance fields. *Proc. CVPR*, 2022.

Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score Jacobian chaining: Lifting pretrained 2d diffusion models for 3D generation. *Proc. CVPR*, 2023a.

Wenjing Wang, Huan Yang, Zixi Tuo, Huiguo He, Junchen Zhu, Jianlong Fu, and Jiaying Liu. Videofactory: Swap attention in spatiotemporal diffusions for text-to-video generation. *arXiv preprint arXiv:2305.10874*, 2023b.

Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *arXiv preprint arXiv:2306.02018*, 2023c.

Yikai Wang, Xinzhou Wang, Zilong Chen, Zhengyi Wang, Fuchun Sun, and Jun Zhu. Vidu4d: Single generated video to high-fidelity 4d reconstruction with dynamic gaussian surfels. *arXiv preprint arXiv:2405.16822*, 2024.

Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3D generation with variational score distillation. *Proc. NeurIPS*, 2023d.

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

Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *Proc. ICLR*, 2023.

Daniel Watson, Saurabh Saxena, Lala Li, Andrea Tagliasacchi, and David J Fleet. Controlling space and time with diffusion models. *arXiv preprint arXiv:2407.07860*, 2024.

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.

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

Zijie Wu, Chaohui Yu, Yanqin Jiang, Chenjie Cao, Fan Wang, and Xiang Bai. Sc4d: Sparse-controlled video-to-4d generation and motion transfer. *arXiv preprint arXiv:2404.03736*, 2024.

Zeqi Xiao, Yifan Zhou, Shuai Yang, and Xingang Pan. Video diffusion models are training-free motion interpreter and controller. *arXiv preprint arXiv:2405.14864*, 2024.

Kevin Xie, Jonathan Lorraine, Tianshi Cao, Jun Gao, James Lucas, Antonio Torralba, Sanja Fidler, and Xiaohui Zeng. LATTE3D: Large-scale amortized text-to-enhanced3D synthesis. *Proc. ECCV*, 2024.

Dejia Xu, Hanwen Liang, Neel P Bhatt, Hezhen Hu, Hanxue Liang, Konstantinos N Plataniotis, and Zhangyang Wang. Comp4d: Llm-guided compositional 4d scene generation. *arXiv preprint arXiv:2403.16993*, 2024a.

Dejia Xu, Weili Nie, Chao Liu, Sifei Liu, Jan Kautz, Zhangyang Wang, and Arash Vahdat. Camco: Camera-controllable 3d-consistent image-to-video generation. *arXiv preprint arXiv:2406.02509*, 2024b.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proc. CVPR*, 2016.

Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. GRM: Large Gaussian reconstruction model for efficient 3D reconstruction and generation. *Proc. ECCV*, 2024c.

Yinghao Xu, Hao Tan, Fujun Luan, Sai Bi, Peng Wang, Jiahao Li, Zifan Shi, Kalyan Sunkavalli, Gordon Wetzstein, Zexiang Xu, et al. DMV3D: Denoising multi-view diffusion using 3D large reconstruction model. *Proc. ICLR*, 2024d.

Zhongcong Xu, Jianfeng Zhang, Jun Hao Liew, Wenqing Zhang, Song Bai, Jiashi Feng, and Mike Zheng Shou. Pv3d: A 3d generative model for portrait video generation. *Proc. ICLR*, 2023.

Hongwei Xue, Tiankai Hang, Yanhong Zeng, Yuchong Sun, Bei Liu, Huan Yang, Jianlong Fu, and Baining Guo. Advancing high-resolution video-language representation with large-scale video transcriptions. In *Proc. CVPR*, 2022.

Qitong Yang, Mingtao Feng, Zijie Wu, Shijie Sun, Weisheng Dong, Yaonan Wang, and Ajmal Mian. Beyond skeletons: Integrative latent mapping for coherent 4d sequence generation. *arXiv preprint arXiv:2403.13238*, 2024a.

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*, 2024b.

Zeyu Yang, Zijie Pan, Chun Gu, and Li Zhang. Diffusion²: Dynamic 3d content generation via score composition of orthogonal diffusion models. *arXiv preprint 2404.02148*, 2024c.

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*, 2024d.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

Junliang Ye, Fangfu Liu, Qixiu Li, Zhengyi Wang, Yikai Wang, Xinzhou Wang, Yueqi Duan, and Jun Zhu. DreamReward: Text-to-3D generation with human preference. *arXiv preprint arXiv:2403.14613*, 2024.

Yuyang Yin, Dejia Xu, Zhangyang Wang, Yao Zhao, and Yunchao Wei. 4dgen: Grounded 4d content generation with spatial-temporal consistency. *arXiv preprint arXiv:2312.17225*, 2023.

Paul Yoo, Jiaxian Guo, Yutaka Matsuo, and Shixiang Shane Gu. DreamSparse: Escaping from Plato's cave with 2D diffusion model given sparse views. *arXiv preprint arXiv:2306.03414*, 2023.

Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019.

Heng Yu, Chaoyang Wang, Peiye Zhuang, Willi Menapace, Aliaksandr Siarohin, Junli Cao, Laszlo A Jeni, Sergey Tulyakov, and Hsin-Ying Lee. 4real: Towards photorealistic 4d scene generation via video diffusion models. *arXiv preprint arXiv:2406.07472*, 2024.

Jason J Yu, Fereshteh Forghani, Konstantinos G Derpanis, and Marcus A Brubaker. Long-term photometric consistent novel view synthesis with diffusion models. In *Proc. ICCV*, 2023a.

Xin Yu, Yuan-Chen Guo, Yangguang Li, Ding Liang, Song-Hai Zhang, and Xiaojuan Qi. Text-to-3D with classifier score distillation. *arXiv preprint arXiv:2310.19415*, 2023b.

Raza Yunus, Jan Eric Lenssen, Michael Niemeyer, Yiyi Liao, Christian Rupprecht, Christian Theobalt, Gerard Pons-Moll, Jia-Bin Huang, Vladislav Golyanik, and Eddy Ilg. Recent trends in 3d reconstruction of general non-rigid scenes. In *Computer Graphics Forum*, 2024.

Yifei Zeng, Yanqin Jiang, Siyu Zhu, Yuanxun Lu, Youtian Lin, Hao Zhu, Weiming Hu, Xun Cao, and Yao Yao. Stag4d: Spatial-temporal anchored generative 4d gaussians. *arXiv preprint arXiv:2403.14939*, 2024.

Bowen Zhang, Tianyu Yang, Yu Li, Lei Zhang, and Xi Zhao. Compress3D: a compressed latent space for 3D generation from a single image. *arXiv preprint arXiv:2403.13524*, 2024a.

Haiyu Zhang, Xinyuan Chen, Yaohui Wang, Xihui Liu, Yunhong Wang, and Yu Qiao. 4diffusion: Multi-view video diffusion model for 4d generation. *arXiv preprint arXiv:2405.20674*, 2024b.

Hao Zhang, Di Chang, Fang Li, Mohammad Soleymani, and Narendra Ahuja. Magicpose4d: Crafting articulated models with appearance and motion control. *arXiv preprint arXiv:2405.14017*, 2024c.

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

Tianyuan Zhang, Hong-Xing Yu, Rundi Wu, Brandon Y Feng, Changxi Zheng, Noah Snavely, Jiajun Wu, and William T Freeman. Physdreamer: Physics-based interaction with 3d objects via video generation. *arXiv preprint arXiv:2404.13026*, 2024d.

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 *Proc. ECCV*, 2022.

Yuyang Zhao, Zhiwen Yan, Enze Xie, Lanqing Hong, Zhenguo Li, and Gim Hee Lee. Animate124: Animating one image to 4d dynamic scene. *arXiv preprint arXiv:2311.14603*, 2023.

Yufeng Zheng, Xueting Li, Koki Nagano, Sifei Liu, Otmar Hilliges, and Shalini De Mello. A unified approach for text-and image-guided 4d scene generation. In *Proc. CVPR*, 2024.

Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. Magicvideo: Efficient video generation with latent diffusion models. *arXiv preprint arXiv:2211.11018*, 2022.

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

A Related Work

Due to space constraints, we summarize related 3D and an extended list of 4D works in the appendix.

3D generation. Limited to single category trainings, early work on 3D generation extends GANs into 3D using a neural renderer as an inductive bias DeVries et al. (2021); Chan et al. (2022); Or-El et al. (2022); Schwarz et al. (2022); Bahmani et al. (2023b), Towards more diverse and flexible generation, CLIP-based supervision Radford et al. (2021) enabled text-based generation and editing of 3D assets Chen et al. (2018); Jain et al. (2022); Sanghi et al. (2022); Jetchev (2021); Gao et al. (2023); Wang et al. (2022). With recent advances in diffusion models, Score Distillation Sampling (SDS) replaces CLIP supervision with diffusion model supervision Poole et al. (2023); Wang et al. (2023d); Lin et al. (2023a); Chen et al. (2023c); Liang et al. (2023); Wang et al. (2023a); Li et al. (2024d); He et al. (2024b); Ye et al. (2024); Liu et al. (2024c); Yu et al. (2023b); Katzir et al. (2024); Lee et al. (2024a); Sun et al. (2024a) for higher quality generation. In order to improve the 3D structure of scenes, another line of work generates multiple views of a scene Lin et al. (2023b); Liu et al. (2023); Shi et al. (2024); Feng et al. (2024a); Liu et al. (2024b); Kim et al. (2023); Voleti et al. (2024b); Höllein et al. (2024). Alternatively, other methods use iterative inpainting for 3D scene generation (Höllein et al., 2023; Shriram et al., 2024). Recent methods lift input images into 3D Chan et al. (2023); Tang et al. (2023); Gu et al. (2023); Liu et al. (2024d); Yoo et al. (2023); Tewari et al. (2023); Qian et al. (2024c); Long et al. (2024); Wan et al. (2024); Szymanowicz et al. (2023) using NeRF (Mildenhall et al., 2020), 3D Gaussian Splatting (Kerbl et al., 2023), or Meshes in combination with diffusion models. Other recent methods Hong et al. (2024); Li et al. (2024b); Xu et al. (2024d;c); Zhang et al. (2024a); Han et al. (2024); Jiang & Wang (2024); Xie et al. (2024); Tang et al. (2024); Tochilkin et al. (2024); Qian et al. (2024a); Szymanowicz et al. (2024b;a) tackle fast feed-forward 3D generation, directly predicting a 3D generation from input images or text. Different from our approach, these methods can only synthesize static scenes.

4D generation. Previous methods also tackle the problem of 4D generation, i.e., generating videos of dynamic 3D scenes from controllable viewpoints, usually from an input text prompt or image. Since the initial work on this topic using large-scale generative models (Singer et al., 2023b), significant improvements in the visual quality and motion quality of generated scenes have been achieved (Ren et al., 2023; Ling et al., 2024a; Bahmani et al., 2024b; Zheng et al., 2024; Gao et al., 2024b; Yang et al., 2024a; Jiang et al., 2024; Bahmani et al., 2024a; Zhang et al., 2024d; Xu et al., 2024a; Miao et al., 2024; Li et al., 2024a). While these methods generate scenes based on text conditioning, other approaches convert an input image or video to a dynamic 3D scene (Ren et al., 2023; Zhao et al., 2023; Yin et al., 2023; Pan et al., 2024; Zheng et al., 2024; Ling et al., 2024a; Gao et al., 2024b; Zeng et al., 2024; Chu et al., 2024; Wu et al., 2024; Yang et al., 2024c; Wang et al., 2024; Feng et al., 2024b; Sun et al., 2024b; Zhang et al., 2024b; Yu et al., 2024; Ren et al., 2024; Lee et al., 2024b; Li et al., 2024c; Van Hoorick et al., 2024; Uzolas et al., 2024; Huang et al., 2024; Chai et al., 2024; Liang et al., 2024; Zhang et al., 2024c). Another line of work (Bahmani et al., 2023a; Xu et al., 2023) extends 3D GANs into 4D by training on 2D videos, however the quality is limited and models are trained on single category datasets. Still, all of these methods are focused on object-centric generation, typically based on 3D volumetric representations. As such, they typically do not incorporate background elements, and overall, they do not approach the level of photorealism demonstrated by the state-of-the-art video generation models used in our technique.

B Quantitative evaluation

We further evaluate all models for the task of single image-to-multiview generation. Furthermore, we provide results for established 2D video generation metrics.

B.1 Multi-view generation

We evaluate our model for image-to-multiview generation. Due to ground truth correspondences for the RealEstate10K (Zhou et al., 2018) dataset, we can condition our model on a given camera trajectory and assess image based metrics, i.e., PSNR, SSIM, and LPIPS. We provide results for the low-resolution base model and the upsampled high-resolution results in Tab. 3.

B.2 Quality metrics

Moreover, we evaluate all models on established image and video generation metrics, namely, FID (Heusel et al., 2017), FVD (Unterthiner et al., 2018), and CLIPSIM (Wu et al., 2021). We provide results for RealEstate10K (Zhou et al., 2018) and MSR-VTT (Xu et al., 2016) in Tab. 4.

It is important to highlight that FID and FVD are not direct measurements of visual quality—these metrics measure similarities between dataset distributions. So while fine-tuning on a dataset will degrade these scores relative to the original model, this degradation is expected because the model is fitting a different data distribution than the one used in the original training run. The same degradation in FID after fine-tuning is observed in the original ControlNet paper (Zhang et al., 2023), where the authors report an increase in FID when comparing the StableDiffusion base model (6.09 FID) to their fine-tuned ControlNet model (15.27 FID; see their Table 3). We believe that joint training with 2D video data may help to alleviate this degradation, and we will explore joint training strategies as part of future work.

B.3 Baseline variants

We provide further variants of MotionCtrl and CameraCtrl adopted to the SnapVideo base model. These variants performed worse than the ones presented in the main paper, hence we mainly show them here for completeness to show the different implementations we explored.

On top of that, we train a variant of MotionCtrl where instead of integrating the camera matrices into the context vector, we integrate them into the output of the cross-attention layer between the latent tokens and patches (as in our approach). We observe degraded camera control compared to our approach. Moreover, we train a variant of CameraCtrl where we incorporate the Plucker embeddings of the CameraCtrl camera encoder into the patches instead of the latents. We observe worse camera control compared to our model. Furthermore, we train a variant of MotionCtrl where we freeze all base model layers similar to CameraCtrl and our approach. Note that MotionCtrl unfreezes the attention layer after injecting camera features into the base model. This experiment highlights that we outperform MotionCtrl independent of training or freezing these attention layers. We show results for camera pose accuracy, multi-view generation, and video quality in Tab. 5, Tab. 6, and Tab. 7 respectively.

B.4 Generalization of camera trajectories

We conduct additional experiments with horizontal and vertical panning, where the camera trajectory is defined by rotation-only camera matrices without any translation. Specifically, we manually construct each trajectory by randomly selecting either the x-, y-, or z-axis and randomly sampling the angular extent of the trajectory (from 0 to 120 degrees). We set the camera translation to the zero vector. We show results in Tab. 8 and observe higher accuracy for our methods. Furthermore, we include results for non-random, user-defined camera trajectories that involve camera movements with significant directional changes including both rotations and translations in Tab. 9. These include following trajectories: rotation around clockwise; rotation around anticlockwise; rotation clockwise without translation; rotation anticlockwise without translation; zoom out, then up translation; translation right, then rotation anticlockwise; translation left, then rotation clockwise; translation left; translation right; translation up; translation down. We observe that our method generalizes to input camera trajectories with variable rotations and translations.

B.5 Experiments with vanilla DiT

Instead of building upon the FIT (Chen & Li, 2023) architecture, we implemented our VD3D method on top of a pre-trained text-to-video DiT model (Peebles & Xie, 2023) in the latent space of the CogVideoX (Yang et al., 2024d) autoencoder. We include these results in Tab. 10 and Tab. 11. Instead of building on top of read attention in FIT, we incorporate the ControlNet conditioning on top of the vanilla attention mechanism of the actual tokens. We observe that the vanilla DiT version further improves quality and camera accuracy on out-of-distribution prompts (MSR-VTT). We believe that our proposed method of spatio-temporal Plucker tokens and aligning them with video patch tokens

Table 3: **Multi-view generation.** We evaluate all models using reference camera trajectories and single-view input images of the RealEstate10K test set. We compute reconstruction metrics based on the subsequent frames for the low-resolution and upsampled high-resolution generations.

Method	Low-resolution			High-resolution		
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)
Base Model	14.74	0.320	0.334	13.23	0.459	0.572
MotionCtrl (U-Net)	15.35	0.387	0.294	13.64	0.473	0.548
CameraCtrl (U-Net)	15.86	0.412	0.266	13.81	0.479	0.540
MotionCtrl	15.07	0.348	0.308	13.42	0.467	0.560
CameraCtrl	14.81	0.327	0.330	13.21	0.456	0.571
Ours	**17.23**	**0.534**	**0.211**	**14.90**	**0.499**	**0.499**
w/o Plucker	14.89	0.346	0.308	13.05	0.455	0.573
w/o ControlNet	14.66	0.313	0.340	13.10	0.450	0.573
w/o weight copy	16.96	0.509	0.220	14.75	0.495	0.504
w/o add context	14.45	0.303	0.368	13.06	0.446	0.588
w/o Plucker context	14.76	0.322	0.318	13.28	0.463	0.579

through a ControlNet-type of conditioning mechanism is agnostic to the transformer architecture and will serve as a starting point for follow-up works.

C Training Details

Compute details. A single training run for the smaller 700M parameter generator takes approximately 1 day on a node equipped with $8\times$ NVIDIA A100 40GB GPUs, connected via NVIDIA NVLink, along with 960 GB of RAM and 92 Intel Xeon CPUs. The larger 4B parameter model was trained on 8 such nodes for 1,5 days, totaling $64\times$ NVIDIA A100 40GB GPUs. In total, we conducted $\approx$150 training runs for the smaller model during the development stage of 4B generator. Consequently, the project's total compute utilization amounted to approximately 2700 NVIDIA A100 40GB GPU-days.

Optimization details. We experiment with two model variants: a smaller generator with approximately 700 million parameters for ablations and initial explorations, and a larger 4 billion parameter model, which we use for the main results in this paper. Both models were trained with a batch size of 256 over 50,000 optimization steps with the LAMB optimizer (You et al., 2019). The learning rate was warmed up for the first 10,000 iterations from 0 to 0.005 and then linearly decreased to 0.0015 over subsequent iterations. For the large model, VD3D contains 230M trainable parameters in total which corresponds to around 5% of the total amount. Since the original video diffusion model is trained in the any-frame-conditioning pipeline (Menapace et al., 2024), we can produce variable camera trajectories from the same starting frame. For text conditioning, we use the T5-11B (Raffel et al., 2020) language model to encode text 1024-dimensional embeddings into 128-length sequences. For training efficiency, they were precomputed for the entire dataset. The rest of the training details have been adopted from (Menapace et al., 2024).

Table 4: **Quality metrics evaluation.** We evaluate all models using text prompts from the RealEstate10K and MSR-VTT test sets respectively.

Method	RealEstate10K			MSR-VTT		
	FID (↓)	FVD (↓)	CLIPSIM (↑)	FID (↓)	FVD (↓)	CLIPSIM (↑)
Base Model	8.22	160.37	0.2677	**3.50**	**141.26**	**0.2774**
MotionCtrl (U-Net)	2.99	61.70	0.2646	16.85	283.12	0.2411
CameraCtrl (U-Net)	2.48	55.64	0.2681	12.33	201.33	0.2505
MotionCtrl	1.50	52.30	0.2708	9.97	183.57	0.2677
CameraCtrl	2.28	66.31	0.2730	8.47	181.90	0.2690
Ours	1.40	42.43	**0.2807**	7.80	165.18	0.2689
w/o Plucker	**1.17**	43.65	0.2715	9.84	152.91	0.2660
w/o ControlNet	3.66	137.06	0.2766	8.34	185.79	0.2674
w/o weight copy	1.38	**42.00**	0.2710	10.09	218.43	0.2647
w/o add context	1.45	44.74	0.2735	9.56	173.43	0.2657
w/o Plucker context	1.54	43.88	0.2724	8.83	168.95	0.2615

Table 5: **Camera pose evaluation with additional baselines.** We evaluate additional variants of the baselines using reference camera trajectories from the RealEstate10K test set. We compute translation and rotation errors based on estimated camera poses from generations using ParticleSfM (Zhao et al., 2022).

Method	RealEstate10K		MSR-VTT	
	TransError (↓)	RotError (↓)	TransError (↓)	RotError (↓)
MotionCtrl (latents)	0.549	0.183	0.650	0.145
CameraCtrl (patches)	0.587	0.197	0.648	0.233
MotionCtrl (frozen)	0.607	0.205	0.678	0.122
Ours	**0.409**	**0.043**	**0.504**	**0.050**

Table 6: **Multi-view generation.** We evaluate additional variants of the baselines using reference camera trajectories and single-view input images of the RealEstate10K test set. We compute reconstruction metrics based on the subsequent frames for the low-resolution and upsampled high-resolution generations.

Method	Low-resolution			High-resolution		
	PSNR (↑)	SSIM (↑)	LPIPS (↓)	PSNR (↑)	SSIM (↑)	LPIPS (↓)
MotionCtrl (latents)	14.67	0.323	0.331	13.23	0.467	0.562
CameraCtrl (patches)	14.42	0.304	0.366	13.04	0.450	0.577
MotionCtrl (frozen)	14.59	0.308	0.340	13.11	0.455	0.573
Ours	**17.23**	**0.534**	**0.211**	**14.90**	**0.499**	**0.499**

Table 7: **Quality metrics evaluation with additional baselines.** We evaluate additional variants of the baselines using text prompts from the RealEstate10K and MSR-VTT test sets respectively.

Method	RealEstate10K			MSR-VTT		
	FID (↓)	FVD (↓)	CLIPSIM (↑)	FID (↓)	FVD (↓)	CLIPSIM (↑)
MotionCtrl (latents)	1.83	77.39	0.2788	10.21	187.42	0.2636
CameraCtrl (patches)	2.57	71.04	0.2703	9.84	184.22	0.2612
MotionCtrl (frozen)	3.53	142.15	0.2772	8.19	165.48	0.2679
Ours	**1.40**	**42.43**	**0.2807**	**7.80**	**165.18**	**0.2689**

Table 8: **Camera pose evaluation for random rotation trajectories.** We evaluate all models using trajectories with randomly selected x-, y-, or z-axis and randomly sampled angular extent of the trajectory. We compute translation and rotation errors based on estimated camera poses from generations using ParticleSfM (Zhao et al., 2022).

Method	RealEstate10K		MSR-VTT	
	TransError (↓)	RotError (↓)	TransError (↓)	RotError (↓)
MotionCtrl	0.396	0.087	0.417	0.120
CameraCtrl	0.381	0.092	0.433	0.138
Ours	**0.202**	**0.037**	**0.265**	**0.044**

Table 9: **Camera pose evaluation for random trajectories with translations and rotations.** We evaluate all models using 11 trajectories for 1000 prompts with trajectories that involve significant directional changes including both rotations and translations. We compute translation and rotation errors based on estimated camera poses from generations using ParticleSfM (Zhao et al., 2022).

Method	RealEstate10K		MSR-VTT	
	TransError (↓)	RotError (↓)	TransError (↓)	RotError (↓)
MotionCtrl	0.451	0.095	0.456	0.146
CameraCtrl	0.369	0.088	0.479	0.135
Ours	**0.236**	**0.041**	**0.258**	**0.050**

Table 10: **Camera pose evaluation with vanilla video DiT backbone.** We incorporate our approach into a pre-trained vanilla DiT model in the latent space of CogVideoX (Yang et al., 2024d). We compute translation and rotation errors based on estimated camera poses from generations using ParticleSfM (Zhao et al., 2022).

Method	RealEstate10K		MSR-VTT	
	TransError (↓)	RotError (↓)	TransError (↓)	RotError (↓)
MotionCtrl	0.518	0.161	0.627	0.148
CameraCtrl	0.532	0.165	0.578	0.220
Ours (FIT DiT)	**0.409**	**0.043**	0.504	0.050
Ours (vanilla DiT)	0.421	0.056	**0.486**	**0.047**

Table 11: **Quality metrics evaluation with vanilla video DiT backbone.** We incorporate our approach into a pre-trained vanilla DiT model in the latent space of CogVideoX (Yang et al., 2024d). We evaluate additional variants of the baselines using text prompts from the RealEstate10K and MSR-VTT test sets respectively.

Method	RealEstate10K			MSR-VTT		
	FID (↓)	FVD (↓)	CLIPSIM (↑)	FID (↓)	FVD (↓)	CLIPSIM (↑)
MotionCtrl	1.50	52.30	0.2708	9.97	183.57	0.2677
CameraCtrl	2.28	66.31	0.2730	8.47	181.90	0.2690
Ours (FIT DiT)	1.40	42.43	0.2807	7.80	165.18	0.2689
Ours (vanilla DiT)	**1.21**	**38.57**	**0.2834**	**6.88**	**137.62**	**0.2790**