MotionCtrl: A Unified and Flexible Motion Controller for Video Generation

Zhouxia Wang*
wzhoux@connect.hku.hk
S-Lab, Nanyang Technological
University
Singapore

Ziyang Yuan*
yuanzy22@mails.tsinghua.edu.cn
Tsinghua University
China

Xintao Wang[†]
xintao.alpha@gmail.com
ARC Lab, Tencent PCG
China

Yaowei Li*
ywl@stu.pku.edu.cn
Peking University
China

Tianshui Chen
tianshuichen@gmail.com
Guangdong University of Technology
China

Menghan Xia
menghanxyz@gmail.com
Tencent AI Lab
China

Ping Luo[†]
pluo@cs.hku.hk
University of Hong Kong
China

Ying Shan
yingsshan@tencent.com
ARC Lab, Tencent PCG
China

(a) Prompt: A cute dog sitting on the green grass.

(b) Prompt: The rose swaying in the wind.

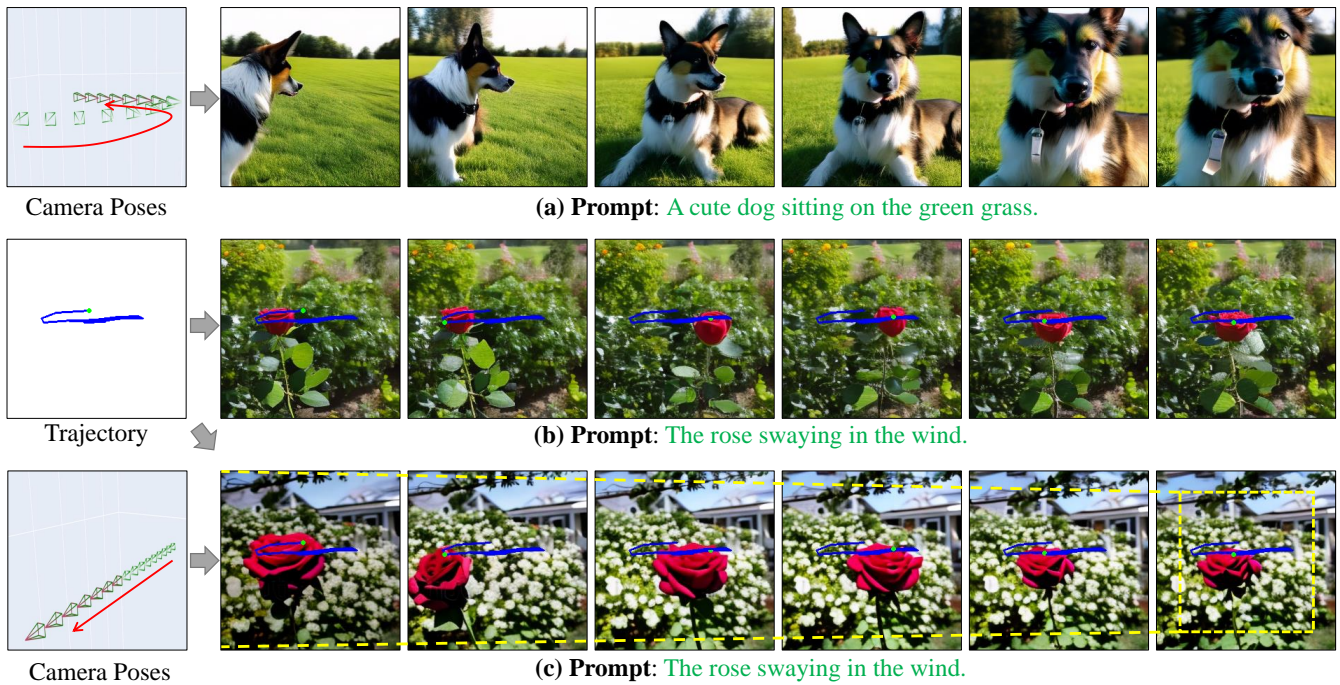**(c) Prompt**: The rose swaying in the wind.

Figure 1: Control Results of MotionCtrl. MotionCtrl is capable of controlling both camera motion and object motion in videos produced by a video generation model. It can also simultaneously control both types of motion within the same video. We highly encourage readers to check our project page for video results, which cannot be well demonstrated by still images.

*Works done while as an intern in ARC Lab, Tencent PCG.
[†]Corresponding author.

ABSTRACT

Motions in a video primarily consist of camera motion, induced by camera movement, and object motion, resulting from object movement. Accurate control of both camera and object motion is essential for video generation. However, existing works either mainly focus on one type of motion or do not clearly distinguish between the two, limiting their control capabilities and diversity. Therefore, this paper presents MotionCtrl, a unified and flexible motion controller for video generation designed to effectively and independently control camera and object motion. The architecture and training strategy of MotionCtrl are carefully devised, taking into account the inherent properties of camera motion, object motion, and imperfect training data. Compared to previous methods, MotionCtrl offers three main advantages: 1) It effectively and independently controls camera motion and object motion, enabling more fine-grained motion control and facilitating flexible and diverse combinations of both types of motion. 2) Its motion conditions are determined by camera poses and trajectories, which are appearance-free and minimally impact the appearance or shape of objects in generated videos. 3) It is a relatively generalizable model that can adapt to a wide array of camera poses and trajectories once trained. Extensive qualitative and quantitative experiments have been conducted to demonstrate the superiority of MotionCtrl over existing methods. Project page: https://wzhouxiff.github.io/projects/MotionCtrl/.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision**.

KEYWORDS

AIGC, video generation, motion control

1 INTRODUCTION

Video generation, such as text-to-video (T2V) generation [Blattmann et al. 2023b; Chen et al. 2023b; He et al. 2022; Ho et al. 2022; Singer et al. 2022; Zhou et al. 2022] aims to produce diverse and high-quality videos that conform to given text prompts. Unlike image generation [Ding et al. 2021; Ramesh et al. 2022, 2021; Rombach et al. 2022; Saharia et al. 2022; Zhou et al. 2021], which focuses on generating a single image, video generation necessitates the creation of consistent and fluent motion among a sequence of generated images. Consequently, motion control plays a significantly crucial role in video generation, yet it has received limited attention in recent research.

In a video, there are primarily two types of motion: global motion induced by camera movements and local motion resulting from object movements (examples are referred to the zoom out camera poses and swaying rose in Fig. 1 (c)). It should be noted that these two motions will be consistently referred to as **camera motion** and

object motion throughout the paper, respectively. However, most previous works related to motion control in video generation either primarily focus on one of the motions or lack a clear distinction between these two types of motion. For instance, AnimateDiff [Guo et al. 2023], Gen-2 [Esser et al. 2023], and PikaLab [pik [n. d.]] mainly execute or trigger camera motion control using independent LoRA [Hu et al. 2021] models or extra camera parameters (such as "-camera zoom in" in PikaLab [pik [n. d.]]). VideoComposer [Wang et al. 2023] and DragNUWA [Yin et al. 2023a] implement both camera motion and object motion using the same conditions: motion vector in VideoComposer [Wang et al. 2023] and trajectory in DragNUWA [Yin et al. 2023a]. The lack of clear distinction between these two motions prevents these approaches from achieving fine-grained and diverse motion control in video generation.

In this paper, we introduce MotionCtrl, a unified and flexible motion controller for video generation, designed to independently control camera and object motion with a unified model. This approach enables fine-grained motion control in video generation and facilitates flexible and diverse combinations of both motion types. However, constructing such a unified motion controller presents significant challenges due to the following two factors. First, camera and object motions differ significantly in terms of movement range and pattern. Camera motion refers to the global transformation of the whole scene across the temporal dimension, which is typically represented through a sequence of camera poses over time. In contrast, object motion involves the temporal movement of specific objects within the scene, and it is usually represented as the trajectory of a cluster of pixels associated with the objects. Second, no existing dataset encompasses video clips that are accompanied by a complete set of annotations, including captions, camera poses, and object movement trajectories. Creating such a comprehensive dataset requires a significant amount of effort and resources.

To address the aforementioned challenges, MotionCtrl deploys a delicately designed architecture, training strategy, and curated datasets. MotionCtrl consists of two modules: the Camera Motion Control Module (CMCM) and the Object Motion Control Module (OMCM), each tailored to handle camera motion and object motion characteristics, respectively. Both CMCM and OMCM function as adapter-like modules integrated into existing video generation models. Specifically, CMCM temporally integrates a sequence of camera poses into the video generation model through its temporal transformers, aligning the global motion of the generated video with the provided camera poses. On the other hand, OMCM spatially incorporates information regarding object movement into the convolutional layers of the video generation model, indicating the spatial positioning of objects in each generated frame. Noted that in this study, we utilize VideoCrafter1 [Chen et al. 2023b], an enhanced version of LVDM [He et al. 2022], as the underlying video generation model, which we refer to as LVDM throughout this paper.

Leveraging the delicately designed architecture reliant on a large-scale pre-trained video diffusion model equipped with adapter-like CMCM and OMCM, we can train these modules separately, thereby mitigating the need for a comprehensive dataset containing videos with annotations of captions, camera poses, and object movement trajectories. Consequently, we achieve MotionCtrl with

two datasets: one contains annotations of captions and camera poses, and another comprises annotations of captions and object movement trajectories. Specifically, we introduce the augmented-Realestate10k dataset, originally annotated with camera movement information. We further enhance this dataset by generating captions using Blip2 [Li et al. 2023], rendering it suitable for training camera motion control in video generation. Additionally, we augment videos sourced from WebVid [Bain et al. 2021] with object movement trajectories synthesized using the motion segmentation algorithm proposed in ParticleSfM [Zhao et al. 2022]. Alongside their original annotated captions, the augmented-WebVid dataset becomes conducive to learning object motion control in video generation. By sequentially and respectively training CMCM and OMCM with these two annotated datasets, our MotionCtrl framework achieves the capability to independently or jointly control camera and object motion within a unified video generation model. This approach enables relatively fine-grained and flexible motion control, empowering users with enhanced control over the generated videos.

Through these delicate designs, MotionCtrl demonstrates superiority over previous methods in three aspects: 1) It independently controls camera and object motion, enabling fine-grained adjustments and a variety of motion combinations, as shown in Fig. 1. 2) It uses camera poses and trajectories for motion conditions, which do not affect the visual appearance, maintaining the objects' natural look in videos. For instance, our MotionCtrl generates a video with a camera motion that closely reflects the reference video, offering a realistic Eiffel Tower, as seen in Fig. 4 (b). In contrast, Video-Composer [Wang et al. 2023] relies on dense motion vectors and mistakenly captures a door's shape of the reference video, resulting in an unnatural Eiffel Tower. 3) MotionCtrl can control a variety of camera movements and trajectories, without the need for fine-tuning each individual camera or object motion.

The main contributions of this work can be summarized as follows: (1) We introduce MotionCtrl, a unified and flexible motion controller for video generation, designed to independently or jointly control camera motion and object motion in generated videos, achieving more fine-grained and diverse motion control. (2) We carefully tailor the architecture and training strategy of MotionCtrl according to the inherent properties of camera motion, object motion, and imperfect training data, effectively achieving fine-grained motion control in video generation. (3) We conduct extensive experiments to demonstrate the superiority of MotionCtrl over previous related methods, both qualitatively and quantitatively.

2 RELATED WORKS

Early research in video generation primarily relied on Generative Adversarial Networks (GANs) or Variational Autoencoders (VAEs) [Saito et al. 2017; Skorokhodov et al. 2022; Tulyakov et al. 2018; Vondrick et al. 2016; Wang et al. 2019,?]. However, in recent years, with the remarkable capacity demonstrated by diffusion models [Ho et al. 2020; Rombach et al. 2022; Saharia et al. 2022] in image generation, video generation research has shifted towards utilizing diffusion models. By further incorporating with text [Blattmann et al. 2023b; Chen et al. 2023b; Guo et al. 2023; He et al. 2022; Ho et al. 2022; Singer et al. 2022; Wang et al. 2023; Zhou et al. 2022]

or image [Blattmann et al. 2023a; Yin et al. 2023b] guidance, diffusion model can generate high-fidelity videos with specific contents. Particularly, the deployment of diffusion models in latent space [Blattmann et al. 2023b; He et al. 2022; Rombach et al. 2022] has significantly enhanced the computational efficiency of video generation, leading to a surge in downstream research centered on diffusion models. MotionCtrl, for instance, aims to leverage diffusion models for controlling motion in generated videos.

In the areas of motion control of generated videos, many existing approaches learn motion by referencing specific or a series of template videos[Guo et al. 2023; Wu et al. 2023b,a; Zhao et al. 2023]. While effective at a specific motion control, these methods typically require training a new model for different templates, which can be limiting. Some efforts aim to achieve more generalized motion control [Chen et al. 2023a; Wang et al. 2023; Yin et al. 2023a]. For instance, VideoComposer [Wang et al. 2023] introduces motion control via extra provided motion vectors, and DragNUWA [Yin et al. 2023a] suggests video generation conditioned on an initial image, provided trajectories, and text prompts. However, the motion control in these methods is relatively broad and fails to fine-grainedly disentangle the camera and object motion within videos.

Different from these works, we propose MotionCtrl, a unified and flexible motion controller that can use either the camera poses and object trajectories or combine these two kinds of guidance to control the motion of generated videos. It enables a more fine-grained and flexible control for video generation.

3 METHODOLOGY

3.1 Preliminary

The Latent Video Diffusion Model (LVDM) [He et al. 2022] aims to generate high-quality and diverse videos guided by text prompts. It employs a denoising diffusion model (U-Net [Ronneberger et al. 2015]) in the latent space for space and time efficiency. Consequently, it constructs a lightweight 3D autoencoder, comprising an encoder $\mathcal{E}$ and a decoder $\mathcal{D}$, to encode raw videos into the latent space and reconstruct the denoised latent features back into videos, respectively. Its denoising U-Net (denoted as ϵ_θ) is constructed with a sequence of blocks that consist of convolutional layers, spatial transformers, and temporal transformers (shown in Fig. 2). It is optimized using a noise-prediction loss:

$$\mathcal{L} = \mathbb{E}_{z_0,c,\epsilon \sim \mathcal{N}(0,I),t} \left[\|\epsilon - \epsilon_\theta(z_t, t, c)\|_2^2 \right], \quad (1)$$

where c represents the text prompt, z_0 is the latent code obtained using $\mathcal{E}$, t ($t \in [0, T]$) denotes the time step, and z_t is the noisy latent features acquired by weighted addition of Gaussian noise ϵ to z_0 using the following formula:

$$z_t = \sqrt{\bar{\alpha}_t} z_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \ \bar{\alpha}_t = \prod_{i=1}^{t} \alpha_t, \quad (2)$$

where α_t is used for scheduling the noise strength based on time step t.

3.2 MotionCtrl

Fig. 2 illustrates the framework of MotionCtrl. To achieve disentanglement between camera motion and object motion, and enable
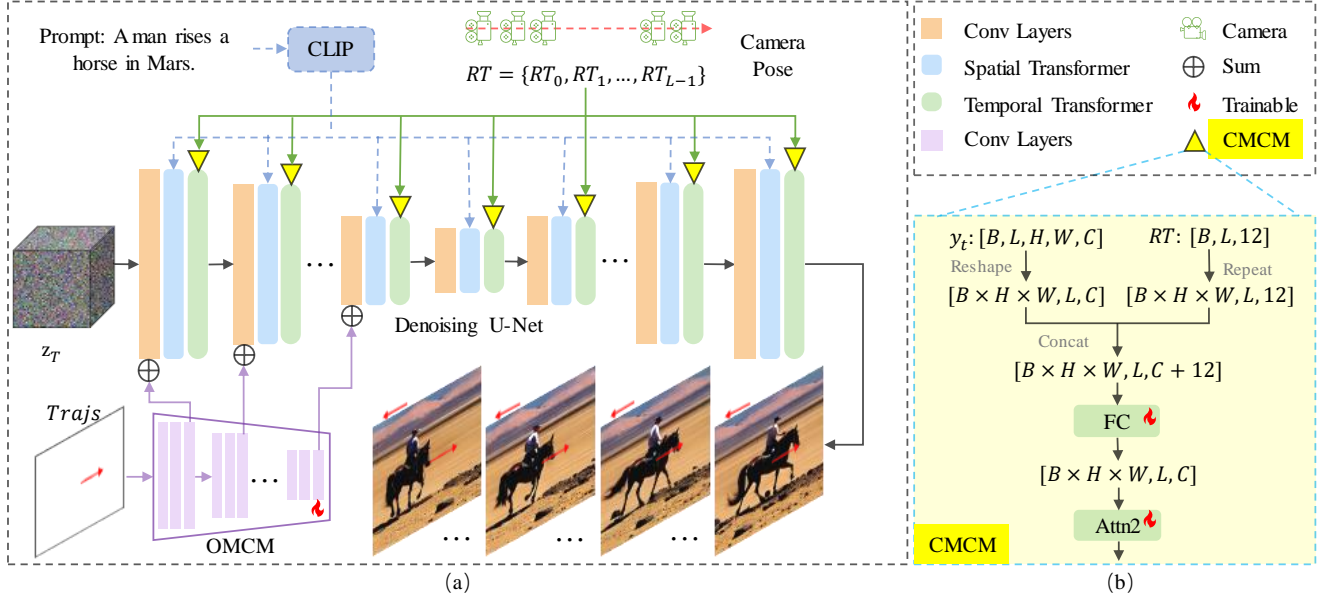
Figure 2: MotionCtrl Framework. MotionCtrl extends the Denoising U-Net structure of LVDM with a Camera Motion Control Module (CMCM) and an Object Motion Control Module (OMCM). As illustrated in (b), the CMCM integrates camera pose sequences RT with LVDM's temporal transformers by appending RT to the input of the second self-attention module and applying a tailored and lightweight fully connected layer to extract the camera pose feature for subsequent processing. The OMCM utilizes convolutional layers and downsamplings to derive multi-scale features from $Trajs$, which are spatially incorporated into LVDM's convolutional layers to direct object motion. Further given a text prompt, LVDM generates videos from noise that correspond to the prompt, with background and object movements reflecting the specified camera poses and trajectories. The resulting video demonstrates the horse moving along its trajectory and meanwhile, the background moves left, consistent with the camera's rightward motion.

independent control of these two types of motion, MotionCtrl comprises two main components: a Camera Motion Control Module (CMCM) and an Object Motion Control Module (OMCM). Taking into account the global property of camera motion and the local property of object motion, CMCM interacts with the temporal transformers in LVDM, while OMCM spatially cooperates with the convolutional layers in LVDM. Furthermore, we employ multiple training steps to adapt MotionCtrl to the absence of training data that contains high-quality video clips accompanied by captions, camera poses, and object movement trajectories. In the following subsections, we will provide a detailed description of CMCM and OMCM along with their corresponding training datasets and training strategies.

3.2.1 Camera Motion Control Module (CMCM).
The CMCM is a lightweight module constructed with several fully connected layers. Since the camera motions are global transformations between frames in a video, CMCM cooperates with LVDM [He et al. 2022] via its temporal transformers. Typically, the temporal transformers in LVDM comprise two self-attention modules and facilitate temporal information fusion between video frames. To minimize the impact on LVDM's generative performance, CMCM only involves the second self-attention module in the temporal transformers. Specifically, CMCM takes a sequence of camera poses $RT = \{RT_0, RT_1, \ldots, RT_{L-1}\}$ as input. In this paper, the camera pose is represented by its 3×3 rotation matrix and 3×1 translation matrix.

Consequently, $RT \in \mathbb{R}^{L \times 12}$, where L denotes the length of the generated video. As depicted in Fig. 2 (b), RT is extended to $H \times W \times L \times 12$ before being concatenated with the output of the first self-attention module in the temporal transformer ($y_t \in \mathbb{R}^{H \times W \times L \times C}$) along the last dimension, where H and W represent the latent spatial size of the generated video, and C is the number of channels in y_t. The concatenated results are then projected back to the size of $H \times W \times L \times C$ using a fully connected layer before being fed into the second self-attention module in the temporal transformer.

3.2.2 Object Motion Control Module (OMCM).
As depicted in Fig. 2, MotionCtrl controls the object motion of the generated video using trajectories ($Trajs$). Typically, a trajectory is represented as a sequence of spatial positions $\{(x_0, y_0), (x_1, y_1), \ldots, (x_{L-1}, y_{L-1})\}$, where $(x_i, y_i), i \in [0, L-1]$ indicates that the trajectory passes through the i_{th} frame at the spatial position (x, y). Particularly, $x \in [0, \hat{W}]$ and $y \in [0, \hat{H}]$, where $\hat{H}$ and $\hat{W}$ are the height and width of z_T, respectively. To explicitly expose the moving speed of the object, we represent $Trajs$ as

$$\{(0, 0), (u_{(x_1, y_1)}, v_{(x_1, y_1)}), \ldots, (u_{(x_{L-1}, y_{L-1})}, v_{(x_{L-1}, y_{L-1})})\}, \text{ where}$$

$$u_{(x_i, y_i)} = x_i - x_{i-1}; v_{(x_i, y_i)} = y_i - y_{i-1}; 0 < i < L. \quad (3)$$

Denoted that the first frame and the other spatial positions in the subsequent frames that the trajectories do not pass are described as $(0, 0)$. Finally, $Trajs \in \mathbb{R}^{L \times \hat{H} \times \hat{W} \times 2}$.

$Trajs$ is injected into LVDM with OMCM, which is highlighted in the purple block of Fig. 2. OMCM consists of multiple convolutional

layers combined with downsampling operations. It extracts multi-scale features from the $Trajs$ and correspondingly adds them to the input of the LVDM's convolutional layers. Drawing inspiration from T2I-Adapter [Mou et al. 2024], the trajectories are only applied to the encoder of the Denoising U-Net to balance the quanlity of the generated video with the ability of object motion control.

3.2.3 Training Strategy and Data Construction. To achieve the control of camera and object motion while generating a video via text prompts, video clips in a training dataset must contain annotations of captions, camera poses, and object movement trajectories. However, a dataset with such comprehensive details is currently unavailable, and assembling one would require considerable effort and resources. To address this challenge, we introduce a multi-step training strategy and train our proposed camera motion control module (CMCM) and object motion control module (OMCM) with distinct augmented datasets tailored to their specific motion control requirements.

Learning the camera motion control module (CMCM). CMCM only requires a training dataset that contains video clips with annotations of captions and camera poses. Considering that Realestate10K [Zhou et al. 2018] contains over 60k videos with relatively clean annotations of camera poses, we take it as our training dataset of CMCM. However, employing Realestate10K in MotionCtrl presents two potential challenges: 1) The diversity of scenes is limited in Realestate10K, primarily from real estate videos, potentially compromising the quality of the generated video; and 2) it lacks captions needed for T2V models.

Regarding the first challenge, we adopt an adapter-like control module (CMCM), with only several new added MLP layers and the second self-attention module of the temporal transformers in LVDM are trainable, and reserving the generation quality of LVDM by freezing most of its parameters. Since the temporal transformers are mainly focus on the learning of global motions, the limited scene diverisity of Realestate10K seldom affects the generation quality of LVDM. This is substantiated by quantitative results presented in Table 2, where the FID [Seitzer 2020] and FVD [Unterthiner et al. 2018] metrics indicate that the video quality generated by our MotionCtrl is on par with the LVDM outcomes.

To address the second challenge, we adopt Blip2 [Li et al. 2023], an image captioning algorithm, to generate captions for each video clip in Realestate10K. Details are in the supplementary materials.

Learning the object motion control module (OMCM). OMCM requires a dataset comprising video clips with captions and object movement trajectories, which is currently lacking in the community. To meet the requirement, we utilize ParticleSfM [Zhao et al. 2022] to synthesize object movement trajectories in WebVid [Bain et al. 2021]. WebVid a large-scale video dataset equipped with captions and commonly used in the T2V generation task. Although ParticleSfM is a structure-from-motion system primarily, it incorporates a trajectory-based motion segmentation module utilized for filtering out dynamic trajectories that affect the production of camera trajectories in a dynamic scene. The dynamic trajectories attained by the motion segmentation module exactly fulfill the requirements of our MotionCtrl and we employ this module to synthesize moving object trajectories for about 243,000 videos in
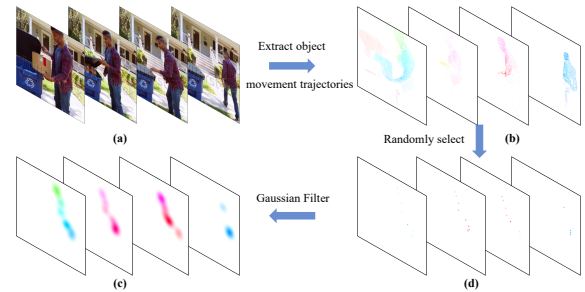

Figure 3: Trajectories for Object Motion Control. ParticleSfM [Zhao et al. 2022] is employed to extract object movement trajectories from video clips, effectively disentangling object motion from camera-induced movement. To circumvent the issues of dense trajectories, which can encode object shapes and are challenging to design at inference, we train the OMCM using sparse trajectories sampled from the dense ones. These sparse trajectories, being too scattered for effective learning, are subsequently refined with a Gaussian filter.

WebVid. An example is illustrated in Fig. 3 (b), where the trajectories predominantly correspond to a moving person. Synthesis details are in the supplementary.

To circumvent the necessity for users to provide dense trajectories as depicted in Fig. 3 (b), which may not be user-friendly, MotionCtrl is required to control the moving objects based on sparse (one or a few) trajectories provided by users. Consequently, our OMCM is trained with $n \in [1, N]$ trajectories (where N represents the maximum number of trajectories for each video) randomly selected from the synthesized dense trajectories (as shown in Fig. 3 (c)). Nevertheless, these selected sparse trajectories tend to be too scattered for effective training. Drawing inspiration from Drag-NUWA [Yin et al. 2023a], we mitigate this issue by applying a Gaussian filter to the sparse trajectories (Fig. 3 (d)) and we initially train the OMCM using dense trajectories before fine-tune it using sparse trajectories.

In this training phase, both LVDM and CMCM are well-trained and frozen, with only the OMCM is trained. This strategy guarantees that OMCM adds the object motion control capabilities with a limited dataset while minimally impacting LVDM and CMCM. Upon the completion of this training phase, giving both camera poses and object trajectories allows for flexible controlling the camera and object motion in the generated video.

4 EXPERIMENTS

4.1 Experiment Settings

4.1.1 Implementation Details. MotionCtrl is built upon the LVDM framework [He et al. 2022]/VideoCraft1 [Chen et al. 2023b], which is trained on 16-frame sequences at a resolution of 256×256. It can be readily adapted to other video generation models with similar structures, such as AnimateDiff [Guo et al. 2023], adhering to the settings specific to each model. Additionally, the maximum number of trajectories N is fixed at 8. Both CMCM and OMCM are optimized using the Adam optimizer [Kingma and Ba 2014] with a batch size of 128 and a learning rate of $1e^{-4}$ across 8 NVIDIA Tesla V100 GPUs. The CMCM typically requires approximately 50,000 iterations to converge. Meanwhile, OMCM undergoes an initial training phase
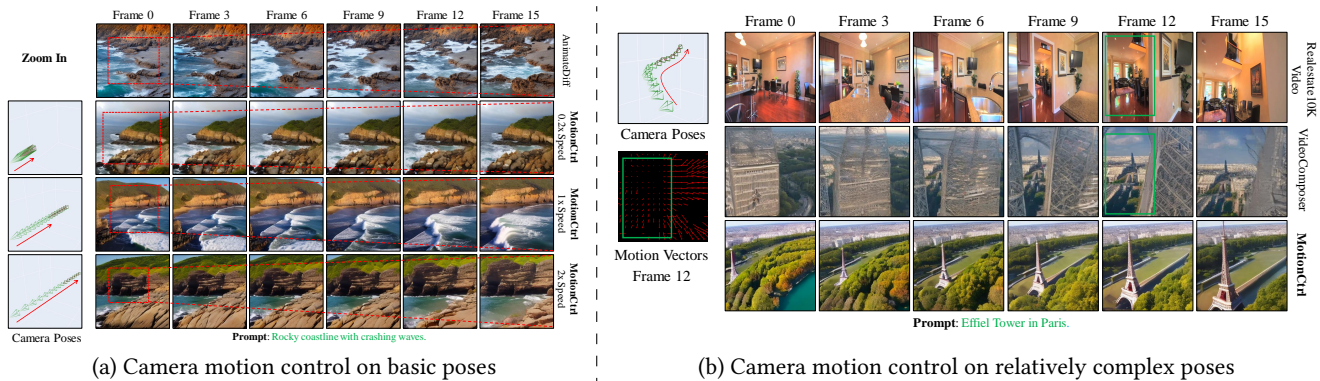
(a) Camera motion control on basic poses

(b) Camera motion control on relatively complex poses

Figure 4: Qualitative Comparisons on Camera Motion Control. (a) Basic Poses: MotionCtrl and AnimateDiff[Guo et al. 2023] effectively execute zooms, but MotionCtrl can adjust to varying camera moving speeds. (b) Relatively Complex Poses: VideoComposer[Wang et al. 2023] uses Realestate10K's raw video for motion vectors, capturing unintended shapes like doors, leading to unnatural results (refer to frame 12). MotionCtrl, however, produces a relatively natural video with motion that closely matches the camera poses.

on dense trajectories for 20,000 iterations, followed by fine-tuning with sparse trajectories for an additional 20,000 iterations.

4.1.2 Evaluation Datasets. (1) Camera motion control evaluation dataset encompasses two types of camera poses: basic camera poses (pan left, pan right, pan up, pan down, zoom in, zoom out, anticlockwise rotation, and clockwise rotation) and relatively complex camera poses[1] obtained from the test set of Realestate10K [Zhou et al. 2018] or synthesized using ParticleSfM [Zhao et al. 2022] on videos from WebVid [Bain et al. 2021] and HD-VILA [Xue et al. 2022]. (2) Object motion control evaluation dataset consists of 283 samples constructed with diverse handcrafted trajectories and prompts. Further details regarding the construction of the evaluation datasets are provided in the supplementary materials.

4.1.3 Evaluation Metrics. (1) The quality of the generated videos is evaluated using Fréchet Inception Distance (**FID**)[Seitzer 2020], Fréchet Video Distance (**FVD**)[Unterthiner et al. 2018], and CLIP Similarity (**CLIPSIM**) [Radford et al. 2021], which measure the visual quality, temporal coherence, and semantic similarity to the text, respectively. Denoted that the reference videos of FID and FVD are 1000 videos from WebVid [Bain et al. 2021]. (2) The efficacy of the camera and object motion control is quantified by computing the Euclidean distance between the predicted and ground truth camera poses and object trajectories, respectively. The camera poses and object trajectories of the predicted videos are extracted using ParticleSfM [Zhao et al. 2022]. We title these two metrics as **CamMC** and **ObjMC**, respectively. 3) We also conduct a user study for subjective quantitative evaluation, the details of which are provided in the supplementary materials due to space limitations.

4.2 Comparisons with State-of-the-Art Methods

To validate the effectiveness of our MotionCtrl in controlling both camera and object motion, we compare it with two leading methods: AnimateDiff [Guo et al. 2023] and VideoComposer [Wang et al.

2023]. AnimateDiff employs 8 separate LoRA [Hu et al. 2021] models to control 8 basic camera motions in videos, such as panning and zooming, while VideoComposer manipulates video motion using motion vectors without differentiating between camera and object movements. Although DragNUWA [Yin et al. 2023a] is relevant to our research, its code is not publicly available, precluding a direct comparison. Moreover, DragNUWA only learns motion control with the trajectories extracted from optical flow, which cannot finegrainedly distinguish the movement between foreground objects and background, limiting its ability to precisely control camera and object motion.

We compare our MotionCtrl with these methods in terms of camera motion and object motion control, and show the capability of our MotionCtrl to flexibly combine the control of camera motion and object motion in video generation. *More comparisons and video comparisons are provided in the supplementary materials.*

4.2.1 Camera Motion Control. We assess camera motion control using basic poses and relatively complex poses. AnimateDiff [Guo et al. 2023] is limited to basic camera poses, while VideoComposer [Wang et al. 2023] handles complex poses by extracting motion vectors from provided videos. The qualitative results are shown in Fig. 4. For basic poses, both MotionCtrl and AnimateDiff can produce videos with forward camera movement, but MotionCtrl can generate camera motion with varying speeds, while AnimateDiff is nonadjustable. Regarding complex poses, where the camera first moves left front and then forward, VideoComposer can mimic the reference video's camera motion using extracted motion vectors. However, the dense motion vectors inadvertently capture object shapes, the door's outline in the reference video (frame 12), resulting in an unnatural-looking Eiffel Tower. MotionCtrl, guided by rotation and translation matrices, generates more natural-looking videos with camera motion close to the reference.

Quantitative results in Table 1 show MotionCtrl's superiority over AnimateDiff and VideoComposer for both basic and relatively complex poses, as reflected by the CamMC score. Additionally,

---

[1]"Complex camera poses" in this work denotes camera movement beyond the basic camera poses. While basic camera poses involve movement in a single straight direction, complex camera poses contain movement in several directions.
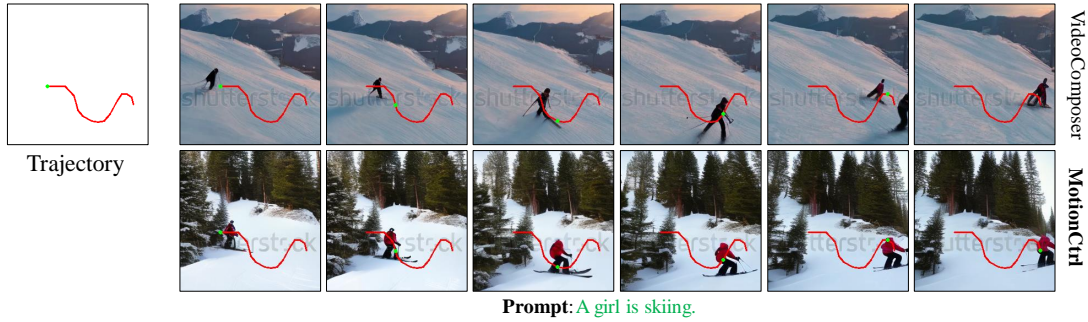
Prompt: A girl is skiing.

Figure 5: Qualitative Comparisons on Object Motion Control. Both VideoComposer and MotionCtrl can generate an object moving along a given trajectory (red curve), but MotionCtrl more precisely follows it in each frame, as indicated by green points.

Table 1: Quantitative Comparisons with AnimateDiff [Guo et al. 2023] and VideoComposer [Wang et al. 2023]. Our MotionCtrl outperforms competing approaches in both camera and object motion control while also excelling at preserving text similarity and the quality of the video generation.

Method	AnimateDiff	VideoComposer	**MotionCtrl**
CamMC ↓ (Basic Poses)	0.0548	-	**0.0289**
CamMC ↓ (Complex Poses)	-	0.0950	**0.0735**
ObjMC ↓	-	36.8351	**28.877**
CLIPSIM ↑	0.2144	0.2214	**0.2319**
FID ↓	157.73	130.97	**124.09**
FVD ↓	1815.88	1004.99	**852.15**

MotionCtrl achieves better text similarity and quality metrics, as measured by CLIPSIM, FID, and FVD.

4.2.2 Object Motion Control. We compare our MotionCtrl with VideoComposer for object motion control, where VideoComposer utilizes motion vectors extracted from trajectories. The qualitative results are shown in Fig. 5. The red curve illustrates the given trajectory, while the green points indicate the expected object locations in the corresponding frame. The visual comparison reveals that MotionCtrl can generate objects whose movements are closer to the given trajectories, whereas VideoComposer's results deviate in certain frames, highlighting MotionCtrl's superior object motion control capability. The quantitative results in terms of ObjMC in Table 1 also demonstrate that MotionCtrl achieves better object motion control than VideoComposer.

4.2.3 Combination of Camera Motion and Object Motion. MotionCtrl can not only control camera and object motion independently within a single video but also perform integrated control of both. As demonstrated in Fig. 1 (b) and (c), when MotionCtrl is applied with only a trajectory, it primarily generates a swaying rose that follows this path. By further introducing zoom-out camera poses, both the rose and the background are animated in accordance with the specified trajectory and camera movements.

More results of MotionCtrl can be found in Fig. 8, supplementary materials, and the demo video.

4.3 Ablation Studies

4.3.1 Integrated Position of Camera Motion Control Module (CMCM).. We test implementing camera motion control by combining camera

Table 2: Ablation of Camera Motion Control. Our Camera Motion Control Module (CMCM), incorporated with the temporal transformers of LVDM [He et al. 2022], effectively controls camera motion and maintains LVDM's video quality.

Method	CamMC ↓	CLIPSIM ↑	FID ↓	FVD ↓
LVDM [He et al. 2022]	0.9010	0.2359	**130.62**	1007.63
Time Embedding	0.0887	0.2361	132.74	1461.36
Spatial Cross-Attention	0.0857	0.2357	153.86	1306.78
Spatial Self-Attention	0.0902	**0.2384**	146.37	1303.58
Temporal Transformer	**0.0289**	0.2355	132.36	**1005.24**

poses with the time embedding, spatial cross-attention, or spatial self-attention module in LVDM. Although such methods have succeeded in other types of controlling [Mou et al. 2024; Zhang et al. 2023], such as sketch and depth, they fail to endow camera control capabilities to LVDM, as evidenced by the CamMC scores in Table 2 and visualized results in Fig. 6. Their CamMC scores are close to the original LVDM. That is because these components primarily focus on spatial content generation, which is insensitive to the camera motion encoded in camera poses. Conversely, incorporating CMCM with LVDM's temporal transformers significantly improves camera motion control, as indicated by a lower CamMC score of 0.0289 in Table 2. Camera motion primarily causes global view transformations over time, and fusing camera poses into LVDM's temporal blocks aligns with this property, enabling effective camera motion control during video generation.

4.3.2 Dense Trajectories v.s. Sparse Trajectories. OMCM is initially trained with dense object movement trajectories extracted via ParticleSfM [Zhao et al. 2022] and then fine-tune with sparse trajectories. We evaluate the effectiveness of this approach by comparing it with training OMCM solely on dense or sparse trajectories. Table 3 and Fig. 7 indicate that training exclusively with dense trajectories yields inferior outcomes, which is attributed to discrepancies between the training and inference phases (sparse trajectories are provided during inference). Though training solely with sparse trajectories shows improvement over the dense-only approach, it still falls short of the hybrid method, since sparse trajectories alone provide limited information. In contrast, dense trajectories offer richer information that accelerates learning, and subsequent fine-tuning with sparse

Table 3: Ablation of Object Motion Control. The Object Motion Control Module (OMCM), when initially trained on dense object movement trajectories and subsequently fine-tuned with sparse trajectories, outperforms versions trained exclusively on either dense or sparse trajectories.

Method	ObjMC ↓	CLIPSIM ↑	FID ↓	FVD ↓
Dense	54.4114	0.2352	175.8622	2227.87
Sparse	34.6937	0.2365	158.5553	2385.39
Dense + Sparse	**25.1198**	**0.2342**	**149.2754**	**2001.57**

trajectories allows OMCM to adjust to the sparsity encountered during inference.

4.3.3 Training Strategy. Given the limitations of the available training dataset, we propose a multi-step training strategy for MotionCtrl, starting with the CMCM using Realestate10K [Zhou et al. 2018], followed by the OMCM with synthesized object movement trajectories. To thoroughly assess our approach, we experiment with reversing the order and training OMCM before CMCM. This sequence does not impact camera motion control, as OMCM components do not participate in CMCM training. However, it leads to a decrease in object motion control performance since the subsequent training of CMCM adjusts parts of LVDM's temporal transformers, disrupting the object motion control adaptation achieved during OMCM's initial training. Thus, our multi-step strategy, though a compromise due to dataset constraints, is deliberately structured to train CMCM before OMCM, ensuring enhanced performance in both camera and object motion control.

4.4 Deploy MotionCtrl on AnimateDiff

We also deploy our MotionCtrl on AnimateDiff [Guo et al. 2023]. Therefore, we can control the motion of the video generated with our adjusted AnimateDiff cooperating with various LoRA [Hu et al. 2021] models in the committee. Visualized results of complex camera motion control and object motion control are in Fig. 9 and Fig. 10. More results are in the supplementary materials.

5 LIMITATIONS

As an initial exploration into controlling camera and object motion within a unified video generation model, MotionCtrl has demonstrated promising and insightful results. However, controlling the camera and object motion in the same video with both complex camera and complex object trajectories requires a careful design of these trajectories to achieve a natural and harmonious outcome, and the success rate is relatively low. Further research is needed to enhance the accuracy of simultaneously controlling camera and object motion in generated videos.
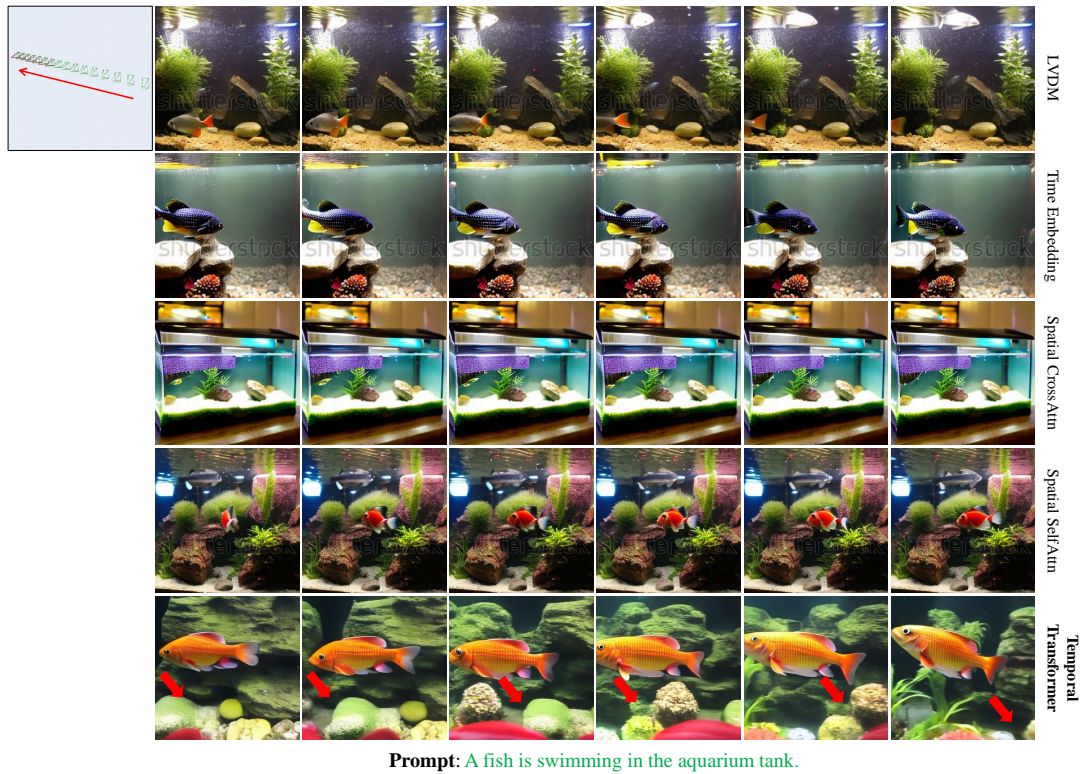
6 CONCLUSION

This paper proposes MotionCtrl, a unified and flexible controller that can independently or combinably control the camera and object motion in a video attained with a video generation model. To achieve this end, MotionCtrl carefully tailors a camera motion control module and object motion control module to adapt to the specific properties of camera motion and object motion and deploys a multi-step training strategy to train these two modules

with delicately augmented datasets. Comprehensive experiments, including qualitative and quantitative evaluations, showcase the superiority of our proposed MotionCtrl in both camera and object motion control.

REFERENCES

[n. d.]. https://www.pika.art/.

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

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

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

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

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

Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. 2021. Cogview: Mastering text-to-image generation via transformers. *NeurIPS* (2021).

Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. 2023. Structure and content-guided video synthesis with diffusion models. In *ICCV*.

Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. 2023. AnimateDiff: Animate Your Personalized Text-to-Image Diffusion Models without Specific Tuning. *arXiv preprint arXiv:2307.04725* (2023).

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

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

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

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685* (2021).

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. 2024. Vbench: Comprehensive benchmark suite for video generative models. (2024).

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. 2024. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *AAAI*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* (2022).

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *ICML*.

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

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

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS* (2022).

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

Maximilian Seitzer. 2020. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid.

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

Ivan Skorokhodov, Sergey Tulyakov, and Mohamed Elhoseiny. 2022. Stylegan-v: A continuous video generator with the price, image quality and perks of stylegan2. In *CVPR*.

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

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

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Generating videos with scene dynamics. *NeuIPS* (2016).

Ting-Chun Wang, Ming-Yu Liu, Andrew Tao, Guilin Liu, Jan Kautz, and Bryan Catanzaro. 2019. Few-shot video-to-video synthesis. *arXiv preprint arXiv:1910.12713* (2019).

Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. 2023. VideoComposer: Compositional Video Synthesis with Motion Controllability. *arXiv preprint arXiv:2306.02018* (2023).

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

Ruiqi Wu, Liangyu Chen, Tong Yang, Chunle Guo, Chongyi Li, and Xiangyu Zhang. 2023a. LAMP: Learn A Motion Pattern for Few-Shot-Based Video Generation. *arXiv preprint arXiv:2310.10769* (2023).

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

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

Shengming Yin, Chenfei Wu, Huan Yang, Jianfeng Wang, Xiaodong Wang, Minheng Ni, Zhengyuan Yang, Linjie Li, Shuguang Liu, Fan Yang, et al. 2023b. Nuwa-xl: Diffusion over diffusion for extremely long video generation. *arXiv preprint arXiv:2303.12346* (2023).

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

Rui Zhao, Yuchao Gu, Jay Zhangjie Wu, David Junhao Zhang, Jiawei Liu, Weijia Wu, Jussi Keppo, and Mike Zheng Shou. 2023. MotionDirector: Motion Customization of Text-to-Video Diffusion Models. *arXiv preprint arXiv:2310.08465* (2023).

Wang Zhao, Shaohui Liu, Hengkai Guo, Wenping Wang, and Yong-Jin Liu. 2022. Particlesfm: Exploiting dense point trajectories for localizing moving cameras in the wild. In *ECCV*.

Daquan Zhou, Weimin Wang, Hanshu Yan, Weiwei Lv, Yizhe Zhu, and Jiashi Feng. 2022. 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. 2018. Stereo magnification: Learning view synthesis using multiplane images. *arXiv preprint arXiv:1805.09817* (2018).

Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. 2021. Lafite: Towards language-free training for text-to-image generation. *arXiv preprint arXiv:2111.13792* (2021).

Prompt: A fish is swimming in the aquarium tank.

Figure 6: The qualitative results of ablation study regarding the integrated position of the Camera Motion Control Module (CMCM) with LVDM [He et al. 2022]. Integrating CMCM of MotionCtrl with the temporal transformers in LVDM significantly improves camera motion control compared to other setups.

Prompt: A man is surfing.

Figure 7: The qualitative results of ablation study "Dense Trajectories v.s. Sparse Trajectories". The model trained with dense trajectories fails to control the object motion in the generated video. Conversely, the model trained on dense trajectories, followed by fine-tuning on sparse trajectories, exhibits superior precision in object motion control compared to the model trained solely on sparse trajectories.

Prompt: A human robot standing on Mars.

Prompt: Two zebras.

Prompt: Seagull, rocks, storm, wind, waves.
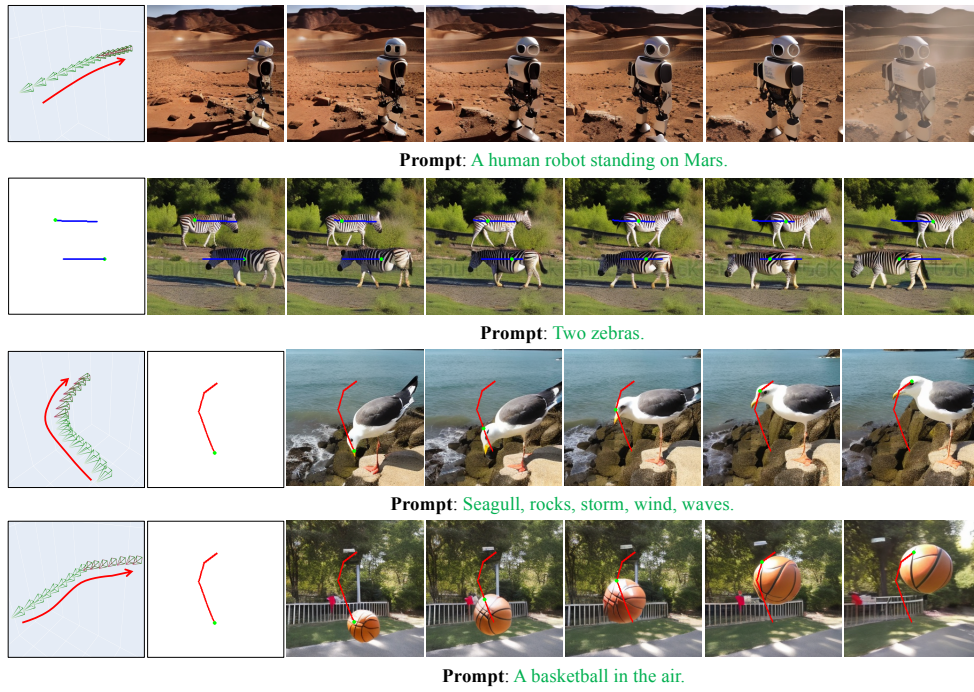
Prompt: A basketball in the air.

Figure 8: More results of MotionCtrl include those controlled by camera poses or object trajectories independently, as well as those controlled with camera poses and object trajectories simultaneously.

Prompt: A girl.

Figure 9: Results of complex camera motion control deployed on AnimateDiff [Guo et al. 2023]

Prompt: : A teddy bear skateboarding.

Figure 10: Results of object motion control deployed on AnimateDiff [Guo et al. 2023].

The supplementary materials provide additional results achieved with our proposed MotionCtrl, along with in-depth analyses. **For a more visual understanding, we strongly recommend readers visit our project page for the video results**. The structure of the supplementary materials is as follows:

- Details of training data construction. (Section A)
- Details of evaluation datasets. (Section B)
- More quantitative and qualitative results. (Section C)
- More Results of MotionCtrl when extended to AnimateDiff [Guo et al. 2023] framework. (Section D)
- More discussions about previous related works. (Section E)

A DETAILS OF TRAINING DATA CONSTRUCTION

Augmented-RealEstate10K. The camera motion control module (CMCM) in MotionCtrl is trained with data augmented from RealEstate10K [Zhou et al. 2018]. RealEstate10K originally contains videos with annotations of camera poses. To adapt it to our MotionCtrl, we further synthesize captions for each video with Blip2 [Li et al. 2023], an image captioning algorithm. Specifically, we extract frames at specific intervals—the first, quarter, half, three-quarters, and final frames of a video. We then use Blip2 to predict their captions. These captions are concatenated to form a comprehensive description for each video clip. With these captions in place, we train the CMCM on RealEstate10K, enabling effective camera motion control in video generation models such as LVDM [He et al. 2022].

Augmented-WebVid. The object motion control module (OMCM) in MotionCtrl is trained with data augmented from WebVid [Bain et al. 2021]. WebVid is a large-scale video dataset equipped with captions and commonly used in the T2V generation task. To adapt it to our MotionCtrl, we further synthesize the object movement trajectories for the videos in WebVid with ParticleSfM [Zhao et al. 2022]. Although ParticleSfM is a structure-from-motion system primarily, it incorporates a trajectory-based motion segmentation module utilized for filtering out dynamic trajectories that affect the production of camera trajectories in a dynamic scene. The dynamic trajectories attained by the motion segmentation module exactly fulfill the requirements of our MotionCtrl and we employ this module to synthesize moving object trajectories required by our MotionCtrl. However, despite its effectiveness, ParticleSfM is not time-efficient, requiring approximately 2 minutes to process a 32-frame video. To mitigate the issue of time efficiency, we randomly select 32 frames from each WebVid video, with a frame skip interval $s \in [1, 16]$, to synthesize the object movement trajectories. This approach yields a total of 243,000 video clips that fulfill the training requirements for the OMCM.

B DETAILS OF EVALUATION DATASETS

In this paper, we construct two evaluation datasets to independently evaluate the efficacy of our proposed MotionCtrl on camera and object motion control, respectively.

Camera Motion Control Evaluation Dataset. This dataset contains a total of **407** samples covering two types of camera poses:

(1) 80 (8 × 10) samples constructed with 8 basic camera pose sequences (pan left, pan right, pan up, pan down, zoom in, zoom out, anticlockwise rotation, and clockwise rotation) and 10 prompts.

(2) 200 (20 × 10) samples constructed with 20 relatively complex camera pose sequences randomly selected from the test set of RealEstate10K [Zhou et al. 2018] and 10 prompts.

(3) 100 samples constructed with 100 relatively complex camera poses of WebVid [Bain et al. 2021] synthesized with ParticleSfM [Zhao et al. 2022] and 100 prompts from VBench [Huang et al. 2024].

(4) 27 samples constructed with 27 relatively complex camera poses of HD-VILA [Xue et al. 2022] synthesized with ParticleSfM and 27 prompts from VBench [Huang et al. 2024].

To provide an intuitive perception of the camera movement, we visualized the 8 basic camera poses and 20 relatively complex camera poses from RealEstate10K [Zhou et al. 2018] in Fig. 11. As described in the manuscript, the term "complex camera poses" as used in this work denotes camera movements beyond the basic camera poses list, encompassing camera turning and self-rotation within the same camera pose.

Object Motion Control Evaluation Dataset. This evaluation dataset contains a total of 283 samples constructed with 74 diverse trajectories and 77 prompts. It should be noted that to verify the effectiveness of MotionCtrl in object motion control, our evaluation dataset pairs one trajectory with several different prompts or one prompt with several different trajectories. To provide an intuitive perception of the handcrafted trajectories, 19 trajectories adopted in the evaluation dataset are depicted in Fig. 12.
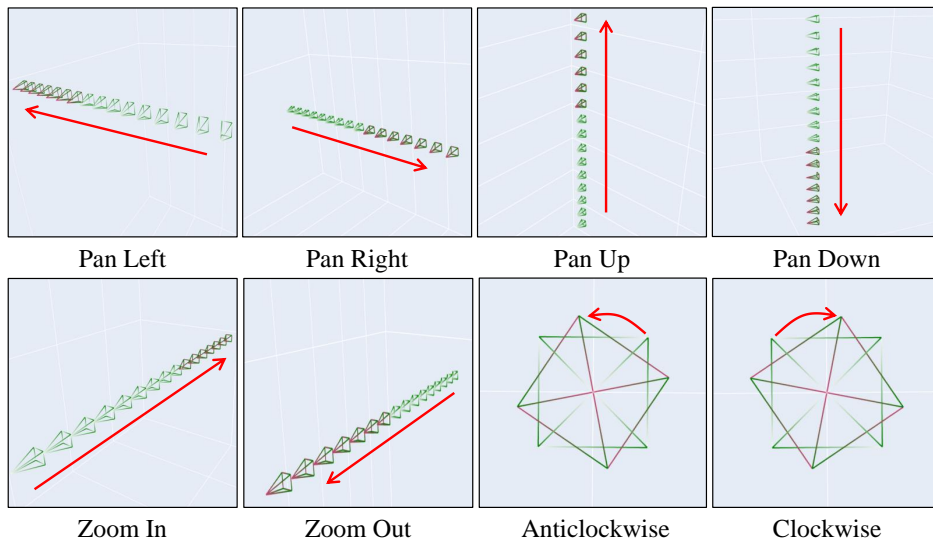
These evaluation datasets will be released.

Please note that the evaluation datasets we have constructed are primarily used for quantitatively assessing the performance of our proposed MotionCtrl in both camera and object motion control in video generation. Our MotionCtrl is capable of handling a wider variety of camera poses and trajectories that are not included in the evaluation datasets.

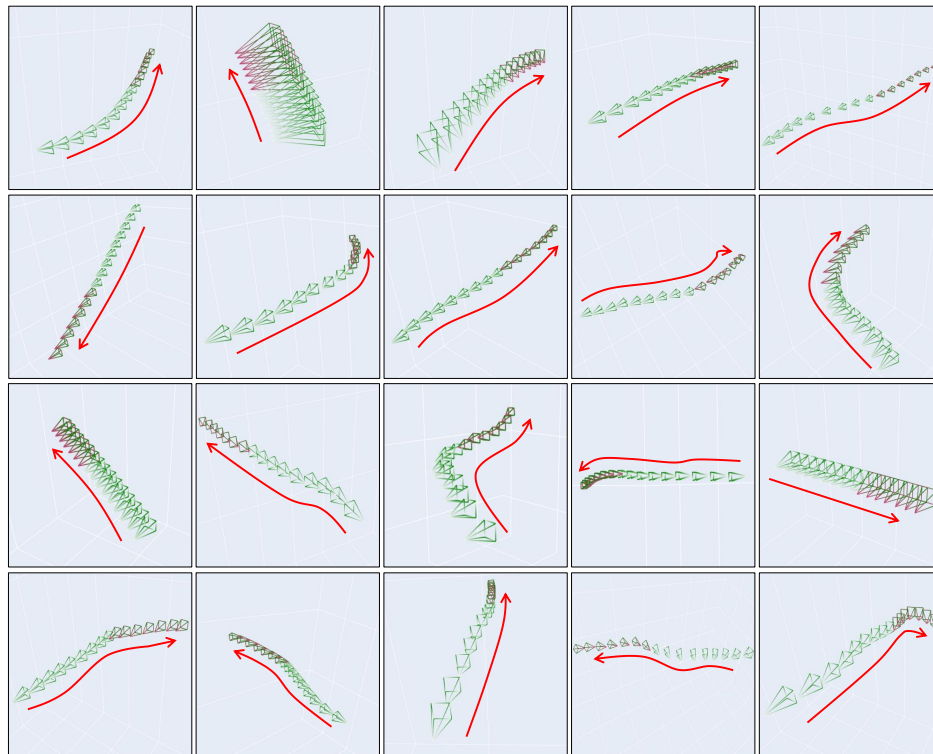## C MORE QUANTITATIVE AND QUALITATIVE RESULTS

C.1 More Quantitative Results

More Quantitative Comparisons on Relatively Complex Camera Motion Control. In the manuscript, the quantitative results of relatively complex camera poses are statistics from all the complex camera poses sourced from RealEstate10K [Zhou et al. 2018], WebVid [Bain et al. 2021], and HD-VILA [Xue et al. 2022]. The statistical results for each dataset are presented in Table 4, demonstrating that our MotionCtrl outperforms VideoComposer [Wang et al. 2023] in both the camera poses extracted from RealEstate10K and those synthesized with ParticleSfM [Zhao et al. 2022] (camera poses of WebVid [Bain et al. 2021] and HD-VILA [Xue et al. 2022]) in terms of camera motion control, text similarity, and generated quality.

User Study. For a more comprehensive evaluation, we conduct a user study involving 34 participants to assess the results of Video-Composer [Wang et al. 2023] and MotionCtrl. The results were generated using object trajectories and relatively complex camera poses covering datasets from RealEstate10K [Zhou et al. 2018], WebVid [Bain et al. 2021], and HD-VILA [Xue et al. 2022]. The assessment included criteria such as Video Quality, Text Similarity,

(a) 8 Basic Camera Poses

(b) 20 Relatively Complex Camera Poses from RealEstate10K Testset

Figure 11: The Camera Motion Control Evaluation Dataset consists of 8 basic camera poses and 20 relatively complex camera poses, with the relatively complex poses being derived from the test set of RealEstate10K. This dataset is utilized to quantitatively assess the effectiveness of our proposed MotionCtrl in controlling a wide range of diverse camera motions in videos generated.

and Motion Similarity. Participants are also asked to express their overall preference for each compared pair. The statistical results in Table 5 demonstrate that over 90 percent of participants preferred

our results in all assessment aspects. Although VideoComposer exhibited good performance in motion control conditioned on motion vectors, its generated videos often appeared unnatural and strange due to the object shapes captured by the motion vectors from the
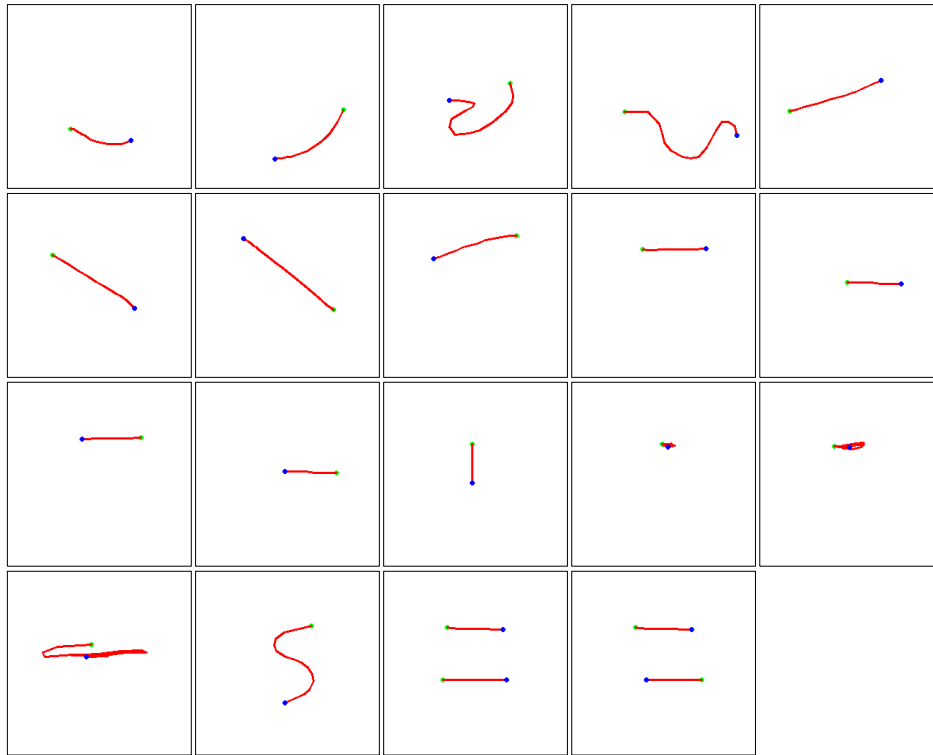
Figure 12: The Object Motion Control Evaluation Dataset encompasses 19 trajectories, where the green and blue points respectively represent the starting and ending points of each trajectory. This dataset is used to quantitatively evaluate the effectiveness of the proposed MotionCtrl in controlling object movements in videos generated.

Table 4: Quantitative Comparisons with VideoComposer [Wang et al. 2023]. Our MotionCtrl performs better in all three sets of relatively complex camera poses from RealEstate10K [Zhou et al. 2018], WebVid [Bain et al. 2021], and HD-VILA [Xue et al. 2022].

Method	RealEstate10K		WebVid		HD-VILA	
	VideoComposer	**MotionCtrl**	VideoComposer	**MotionCtrl**	VideoComposer	**MotionCtrl**
CamMC ↓	0.1073	**0.0840**	0.0702	**0.0589**	0.0953	**0.0499**
CLIPSIM ↑	0.2219	**0.2324**	0.2147	**0.2268**	0.2429	**0.2473**
FID ↓	134.97	**130.29**	106.89	**102.13**	190.54	**159.52**
FVD ↓	1045.82	**934.37**	733.09	**612.84**	1709.59	**1129.40**

reference video. Consequently, users showed a stronger preference for our relatively natural results.

Table 5: User Study. Compared to the results generated with VideoComposer [Wang et al. 2023], our MotionCtrl achieved more preference in all assessment aspect.

Method	VideoComposer	**MotionCtrl**
Quality ↑	0.0628	**0.9372**
TextSimilarity ↑	0.0772	**0.9228**
MotionSimilarity ↑	0.086	**0.9140**
OverallPreference ↑	0.0739	**0.9261**

C.2 More Qualitative Results

More Qualitative Comparisons with VideoComposer. We present additional qualitative results comparing VideoComposer [Wang et al. 2023] and our proposed MotionCtrl on relatively complex camera and object trajectories in Fig. 13 and Fig. 14, respectively. These results suggest that MotionCtrl outperforms VideoComposer in both camera and object motion control in generated videos. Furthermore, MotionCtrl's generated videos exhibit higher quality and its generated content is better aligned with the prompts.

More of MotionCtrl. In this section, we present additional results of MotionCtrl, focusing on camera motion control, object motion control, and combined motion control. **Notably, all results are obtained using the same trained MotionCtrl model, without**

Prompt: Coastline, rocks, storm, wind, waves, lightning.

Prompt: A plant on a soil.
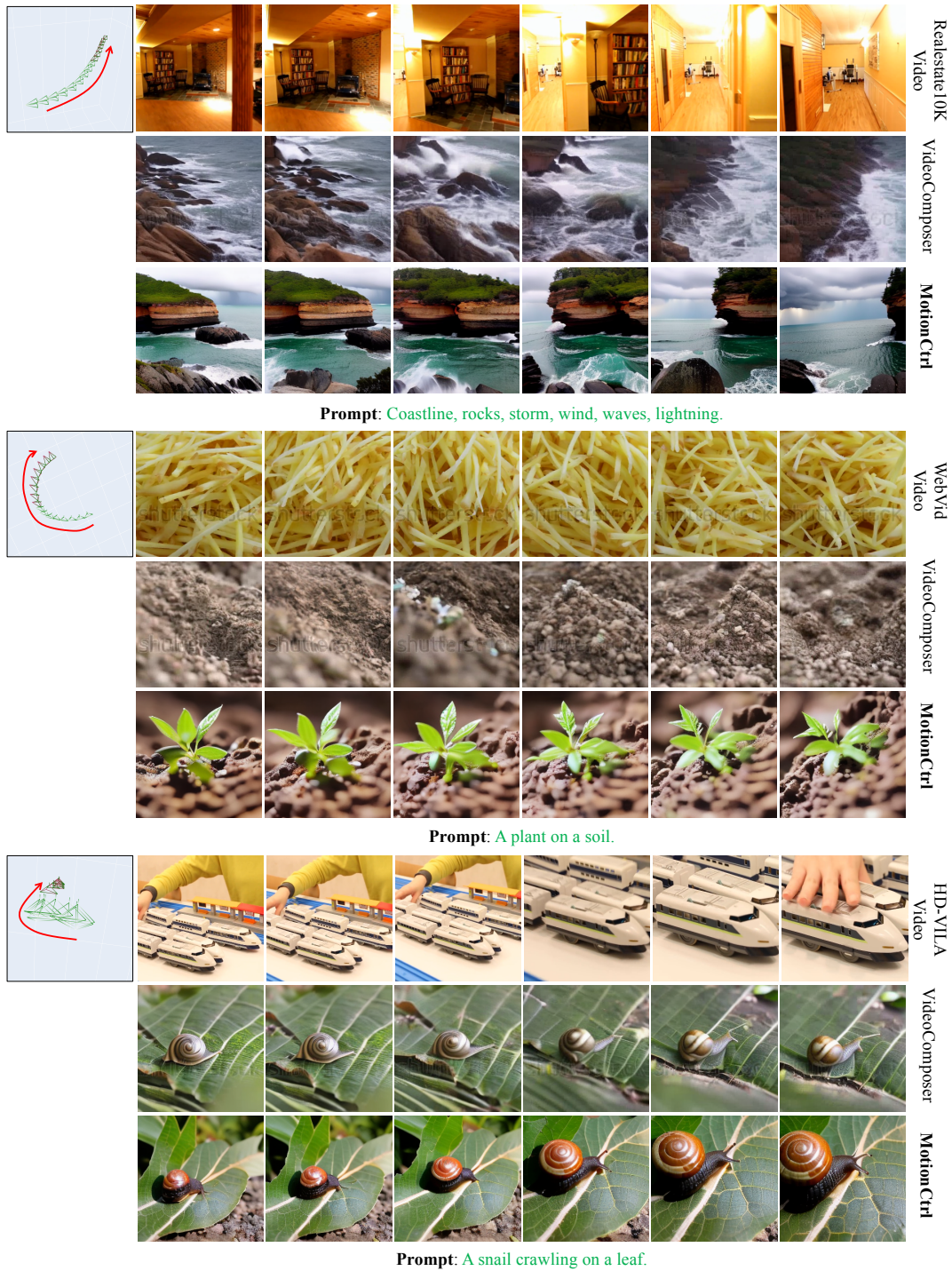
Prompt: A snail crawling on a leaf.

Figure 13: More Qualitative Comparisons with VideoComposer [Wang et al. 2023] on Camera Motion Control. The generated videos of MotionCtrl can better follow the camera poses, whether from RealEstate10K [Zhou et al. 2018] or those synthesized with ParticleSfM [Zhao et al. 2022] on videos of WebVid [Bain et al. 2021] and HD-VILA [Xue et al. 2022]. Moreover, the results achieved with MotionCtrl exhibit higher quality.

Prompt: A lizard on a bamboo.

Prompt: Cow in a field.

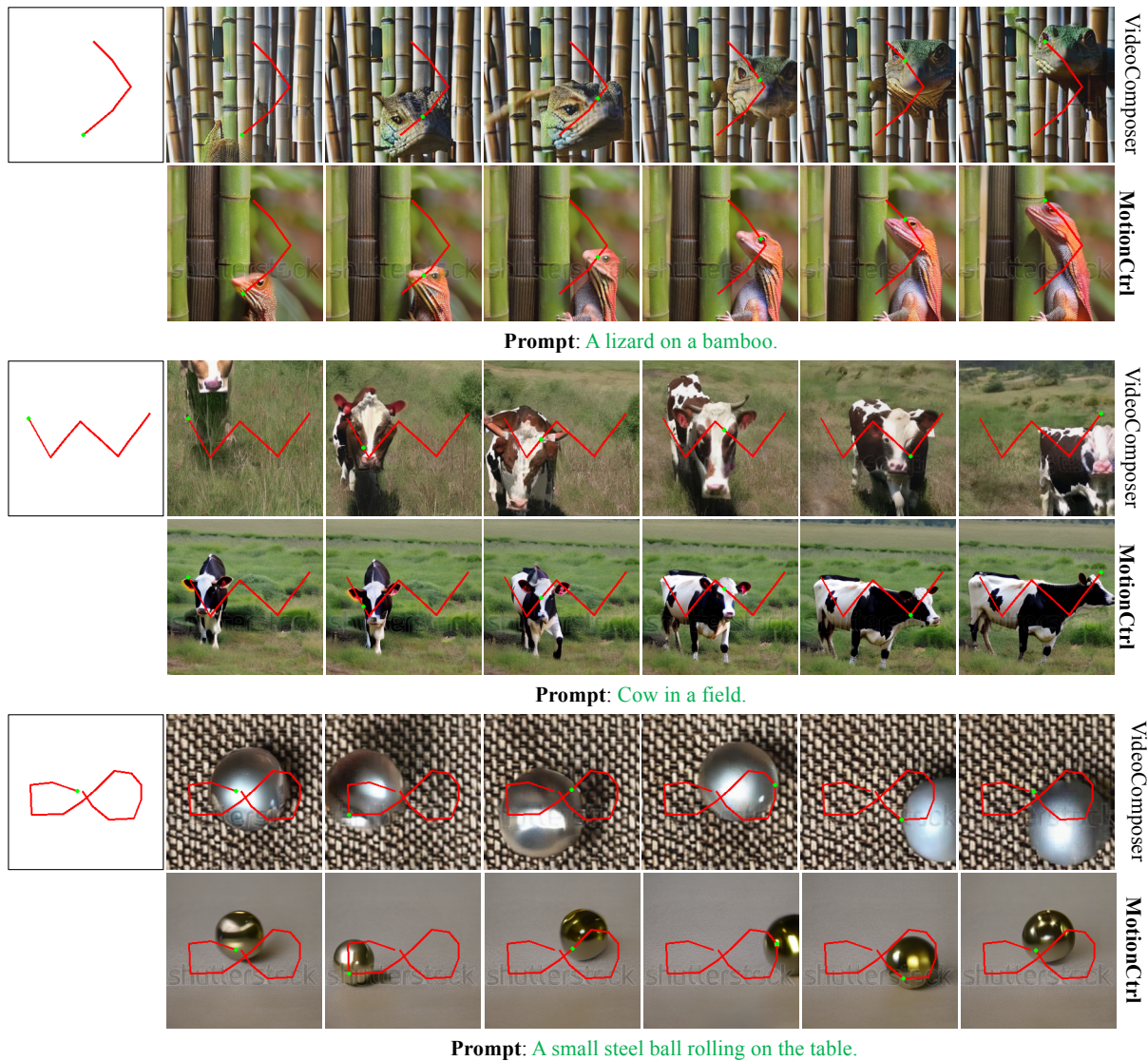**Prompt**: A small steel ball rolling on the table.

Figure 14: More Qualitative Comparisons with VideoComposer [Wang et al. 2023] on Object Motion Control. The generated videos of MotionCtrl exhibit a superior ability to follow the trajectories in each frame and achieve higher overall quality.

the need for extra fine-tuning for different camera poses or trajectories.

Specifically, Fig. 17 illustrates the outcomes of camera motion control of MotionCtrl guided by 8 basic camera poses, including pan up, pan down, pan left, pan right, zoom in, zoom out, anticlockwise rotation, and clockwise rotation. These poses are visualized in Fig. 11 (a). This demonstrates the capability of our MotionCtrl model to integrate multiple basic camera motion controls in a unified model, contrasting with the AnimateDiff model [Guo et al. 2023] which requires a distinct LoRA model [Hu et al. 2021] for each camera motion.

Fig. 15 showcases the results of camera motion control using MotionCtrl, which is guided by relatively complex camera poses. **These complex camera poses are distinct from basic camera**

poses, as they include elements of camera turning or self-rotation within the same camera pose sequence. The results demonstrate that, given a sequence of camera poses, our MotionCtrl can generate natural videos. The content of these videos aligns with the text prompts, and the camera motion corresponds to the provided complex camera poses.

Fig. 18 presents the results of object motion control using MotionCtrl, guided by specific trajectories. When given the same trajectories and different text prompts, MotionCtrl can generate videos featuring different objects, but with identical object motion.

Fig. 16 provides the results of combining both the camera motion control and object motion control. With the same trajectory but different camera poses, the horse in the generated videos has a different performance.

Prompt: A cute cat lying on the floor.
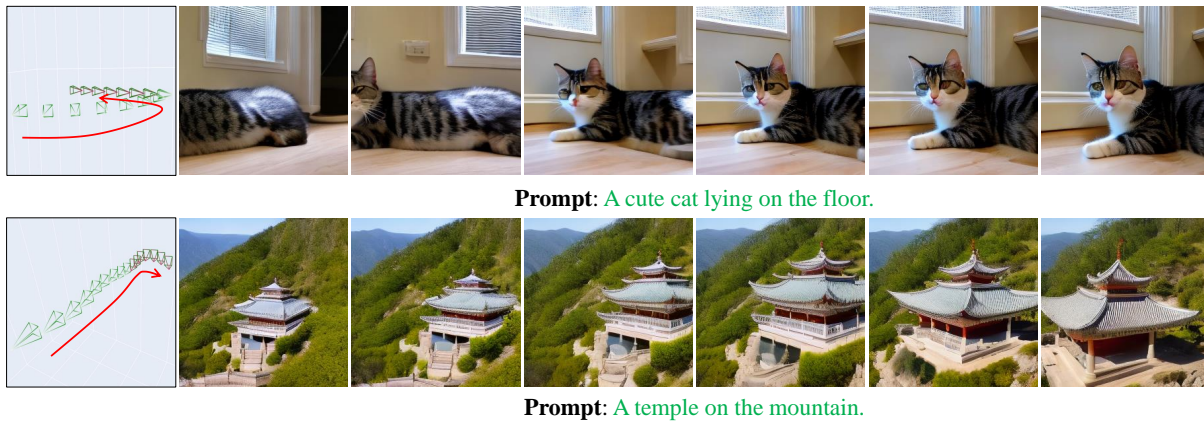

Prompt: A temple on the mountain.

Figure 15: The results of our proposed MotionCtrl deployed on LVDM [He et al. 2022], guided by relatively complex camera poses. Unlike basic camera poses, which only involve simple directional movements, these complex camera poses incorporate elements of camera turning or self-rotation within the same camera pose sequence. The camera motion in the generated videos closely follows the guided camera poses, while the generated content aligns with the text prompts.

Zoom In

Pan Left

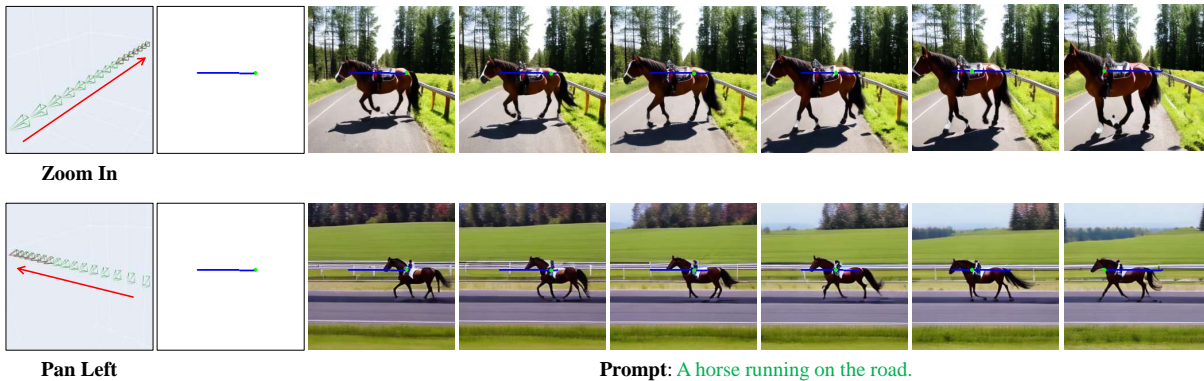**Prompt**: A horse running on the road.

Figure 16: The result of combining camera motion and object motion control of MotionCtrl deployed on LVDM [He et al. 2022]. With the same trajectory but different camera poses, the horse in the generated videos has a different performance.

D MORE RESULTS OF MOTIONCTRL DEPLOYED ON AIMATEDIFF [Guo et al. 2023]

We also deploy our MotionCtrl on AnimateDiff [Guo et al. 2023]. Therefore, we can control the motion of the video generated with our fine-tuned AnimateDiff cooperating with various LoRA [Hu et al. 2021] models in the committee. Results of relatively complex camera motion control and object motion control are in the manuscripts and we provide the results of basic camera motion control here: Fig. 19 and Fig. 20. These results are generated with our MontionCtrl cooperating with different LoRA models provided by in CIVITAI [?]. They demonstrate that our the generalization of MotionCtrl that can be adapted to different video generation models.
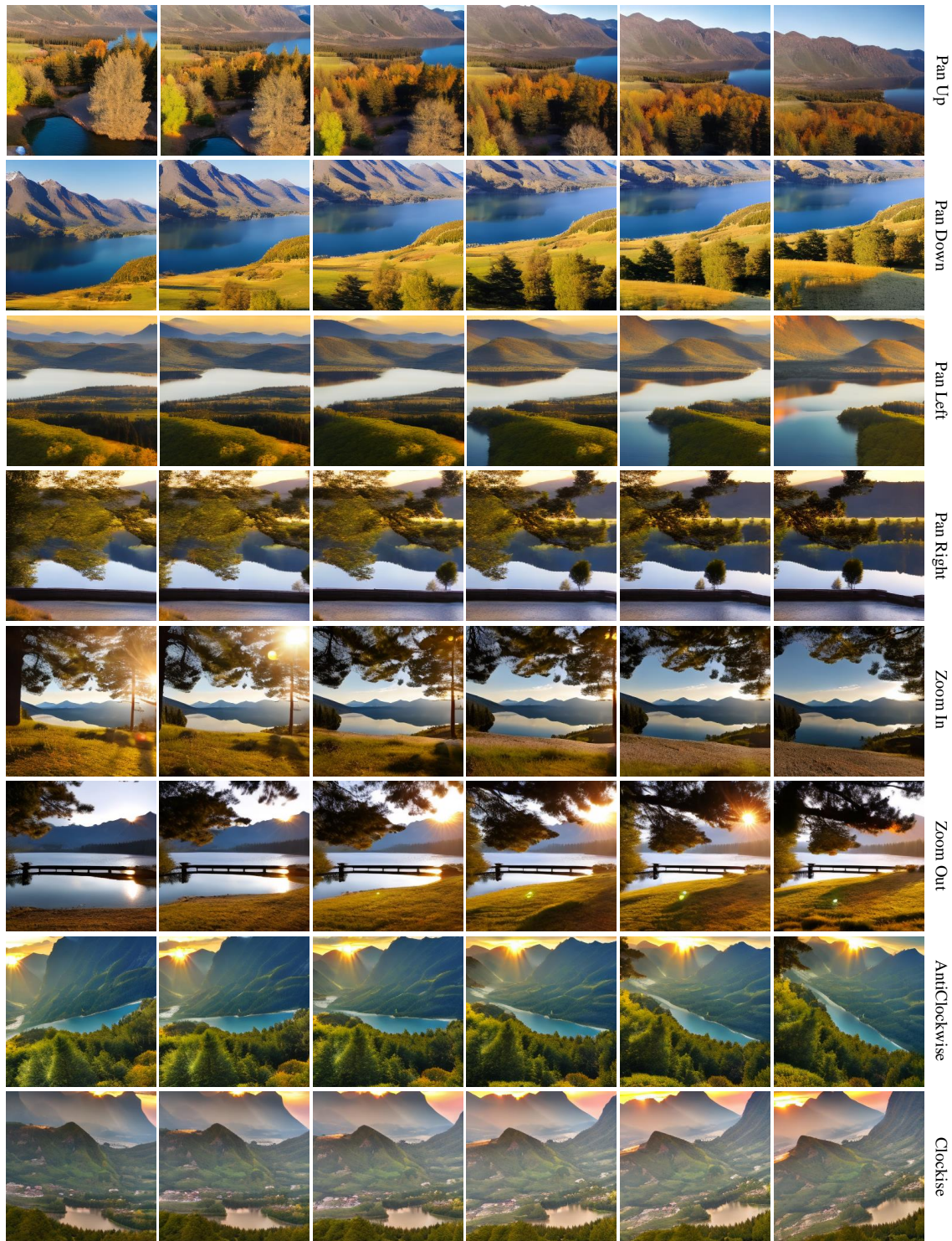
E MORE DISCUSSIONS ABOUT THE RELATED WORKS

To further illustrate the advantages of our proposed MotionCtrl, we've conducted a comparative analysis with previous related works. The comparisons are detailed in Table 6. Models such as AnimateDiff[Guo et al. 2023] (refers to the motion control LoRA models provided by AnimateDiff), Tune-a-video[Wu et al. 2023b], LAMP[Wu et al. 2023a], and MotionDirector[Zhao et al. 2023] implement motion control by extracting motion from one or multiple template videos. This approach necessitates the training of distinct models for each template video or template video set. Moreover, the motions these methods learned are solely determined by the template video(s), and they fail to differentiate between camera motion and object motion. Similarly, MotionDirector[Zhao et al. 2023] and VideoComposer[Wang et al. 2023], despite achieving motion control with a unified model guided by motion vectors and trajectories respectively, do not distinguish between camera motion and object motion. In contrast, our proposed MotionCtrl, utilizing

Table 6: Differences between our proposed MotionCtrl and related works. Unlike AnimateDiff [Guo et al. 2023] (which refers to the motion control LoRA model provided by AnimateDiff), Tune-a-video [Wu et al. 2023b], LAMP [Wu et al. 2023a], and MotionDirector [Zhao et al. 2023] that implement motion control by extracting motion from one or a series of template videos and require different models for different template videos, our proposed MotionCtrl uses a unified model. Besides, the motions learned by these methods are determined by the template video(s) and they do not distinguish between camera motion and object motion. On the other hand, although MotionDirector [Zhao et al. 2023] and VideoComposer [Wang et al. 2023] achieve motion control with a unified model guided by motion vectors and trajectories, respectively, they also do not distinguish between camera motion and object motion. In contrast, our proposed MotionCtrl, with a unified model, can independently and flexibly control the camera motion and object motion of the generated video, guided by camera poses and trajectories, respectively.

Method	Require Fine-tuning	Motion sources	Distinguish Camera & Object Motion
AnimateDiff [Guo et al. 2023]	✓	template videos	✗
Tune-a-video [Wu et al. 2023b]	✓	template video	✗
LAMP [Wu et al. 2023a]	✓	template videos	✗
MotionDirector [Zhao et al. 2023]	✓	template videos	✗
VideoComposer [Wang et al. 2023]	✗	motion vectors	✗
DragNUWA [Yin et al. 2023a]	✗	trajectories	✗
MotionCtrl (Ours)	✗	camera poses & trajectories	✓

a unified model, can independently and flexibly control a wide range of camera and object motions in the generated videos. This is achieved by guiding the model with camera poses and trajectories respectively, offering a more fine-grained control over the video generation process.

Prompt: A landscape with mountains and lake at sunrise.

Figure 17: The results of our proposed MotionCtrl deployed on LVDM [He et al. 2022], guided by 8 basic camera poses: pan up, pan down, pan left, pan right, zoom in, zoom out, anticlockwise rotation, and clockwise rotation (The visualization of these camera poses can be seen in Fig. 11 (a)). It's important to note that all results are achieved using the same MotionCtrl model, without the need for extra fine-tuning for different camera poses.
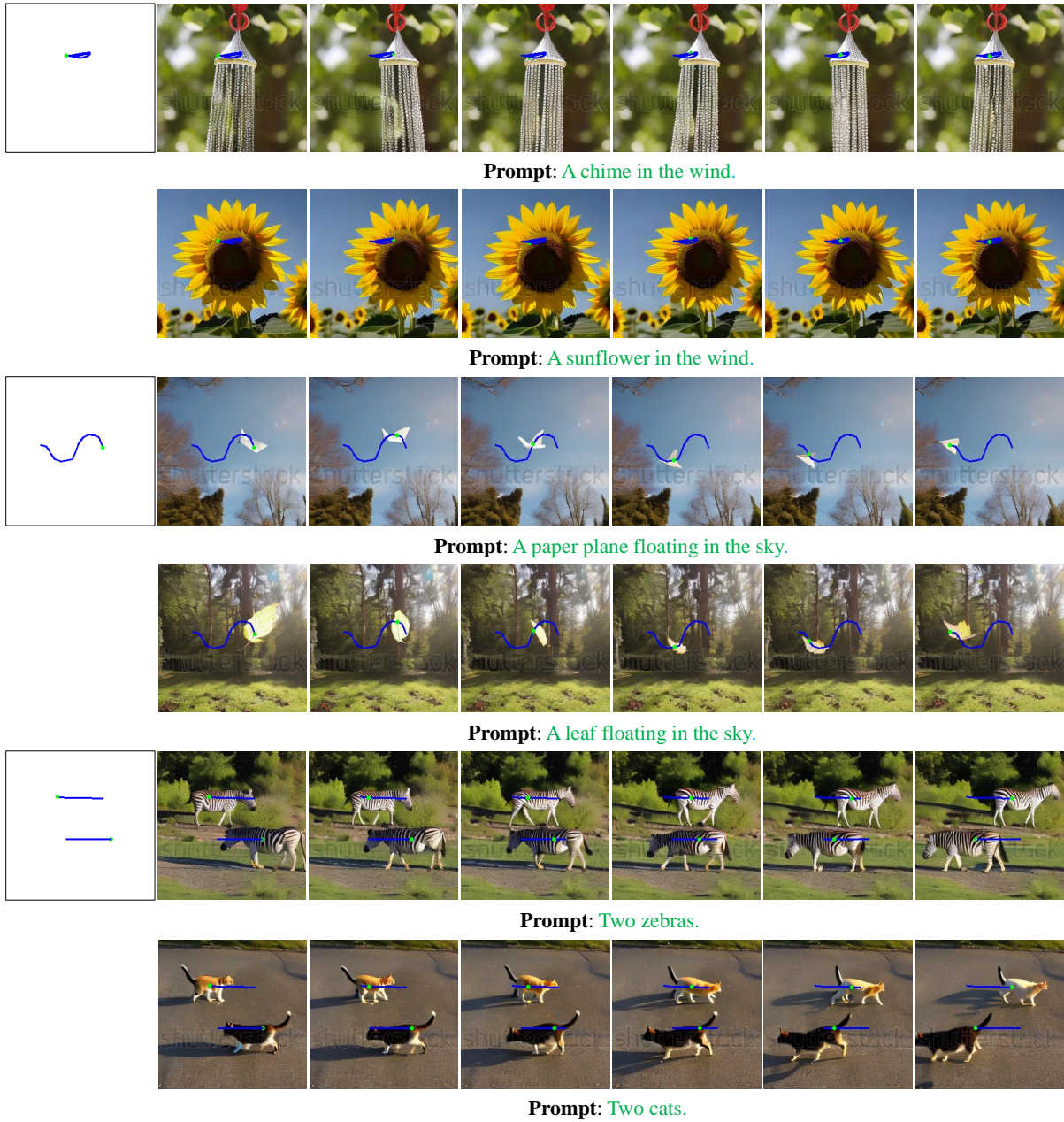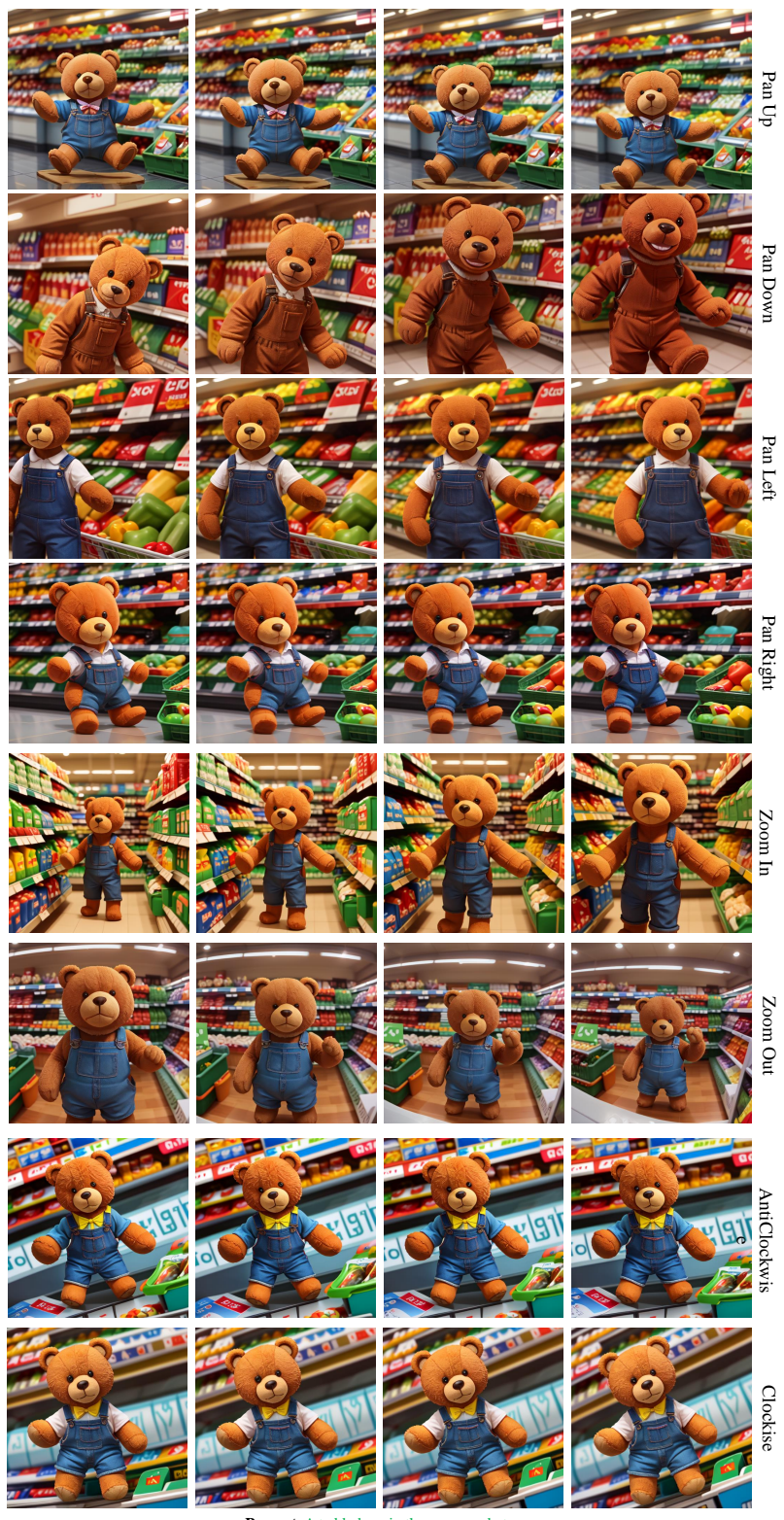
Prompt: A chime in the wind.

Prompt: A sunflower in the wind.

Prompt: A paper plane floating in the sky.

Prompt: A leaf floating in the sky.

Prompt: Two zebras.

Prompt: Two cats.

Figure 18: The result of our proposed MotionCtrl deployed on LVDM [He et al. 2022], guided with trajectories. The green points in the trajectories indicate the starting points. Given the same trajectories, our model can generate different objects in accordance with the text prompts, maintaining the same object motion. When multiple trajectories are present in the same video, our model is capable of simultaneously controlling the motion of different objects within the same generated video.

Prompt: A teddy bear in the supermarket.

Figure 19: The camera motion control results of MotionCtrl deployed on AnimateDiff [Guo et al. 2023]. They are guided with 8 basic camera poses.

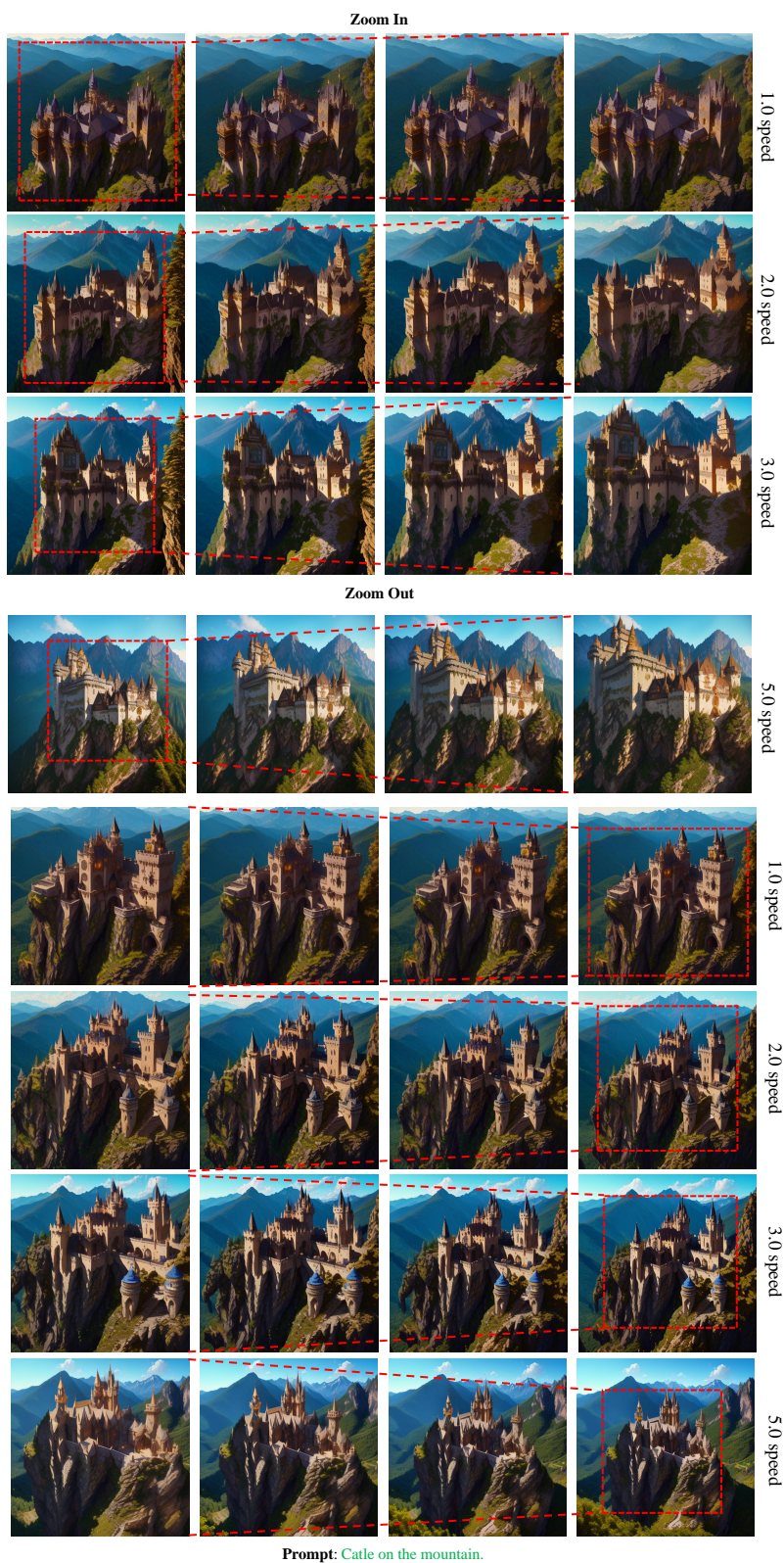**Prompt**: Catle on the mountain.

Figure 20: The camera motion control results of MotionCtrl deployed on AnimateDiff [Guo et al. 2023]. Our MotionCtrl can not only control the camera motion of the generated videos but also their motion speed.